FUJIFILM 92828060

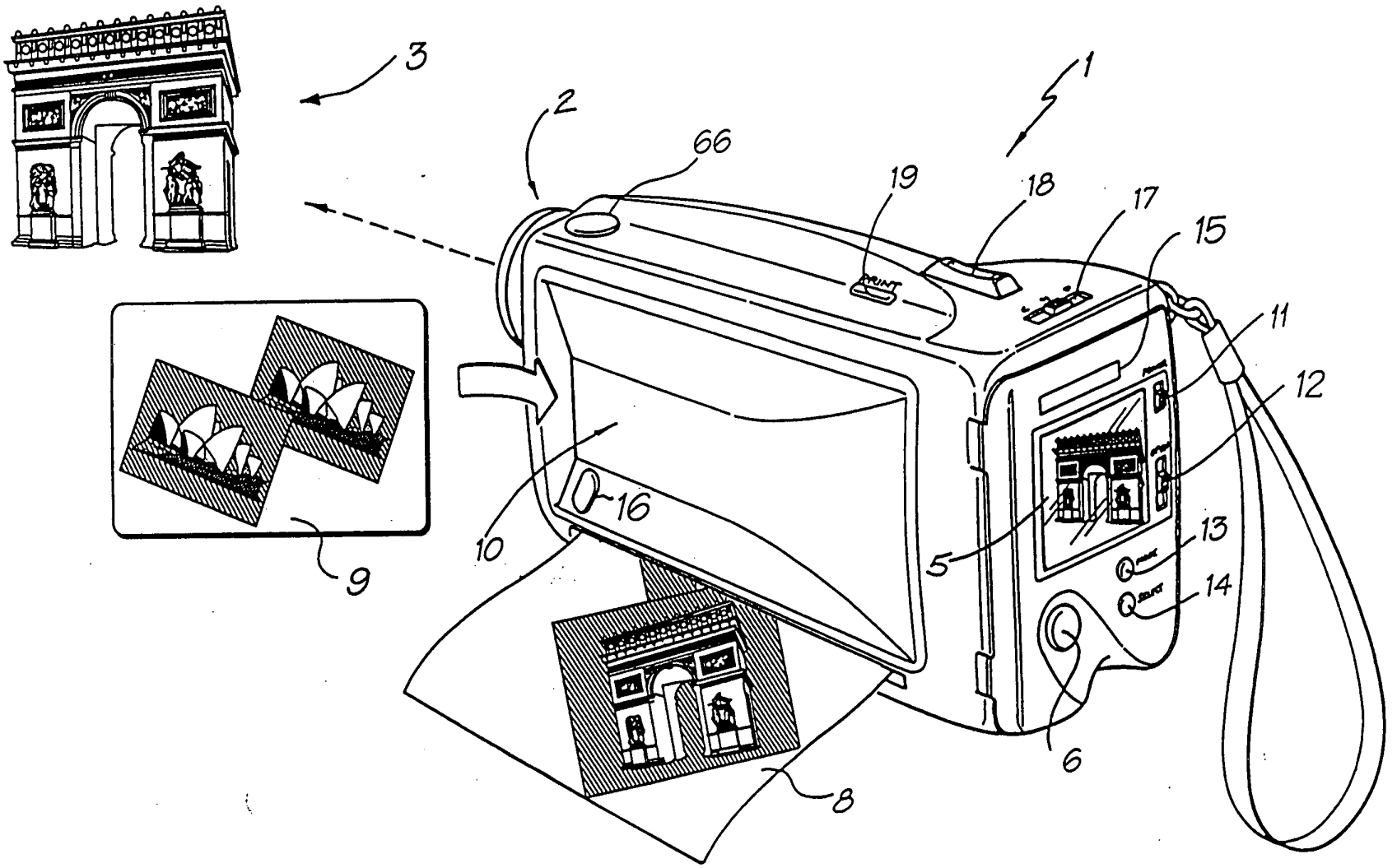


FIG. 1

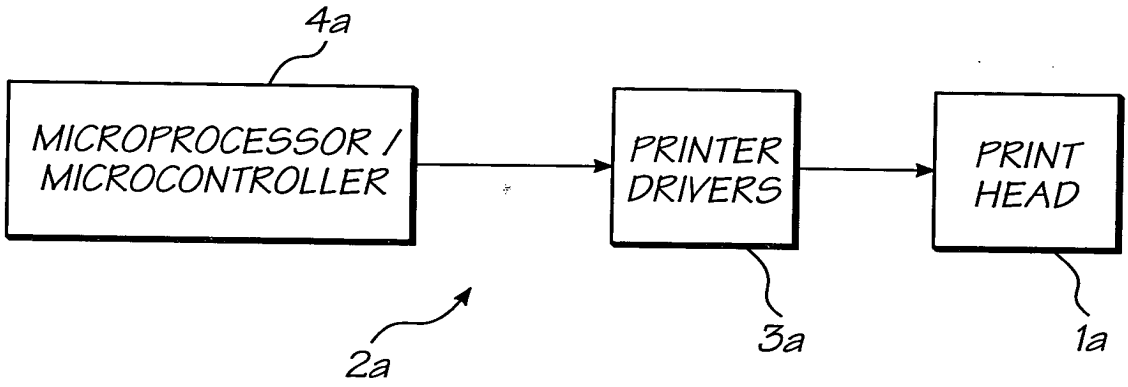


FIG. 1A
(PRIOR ART)

PCT/US2003/039819

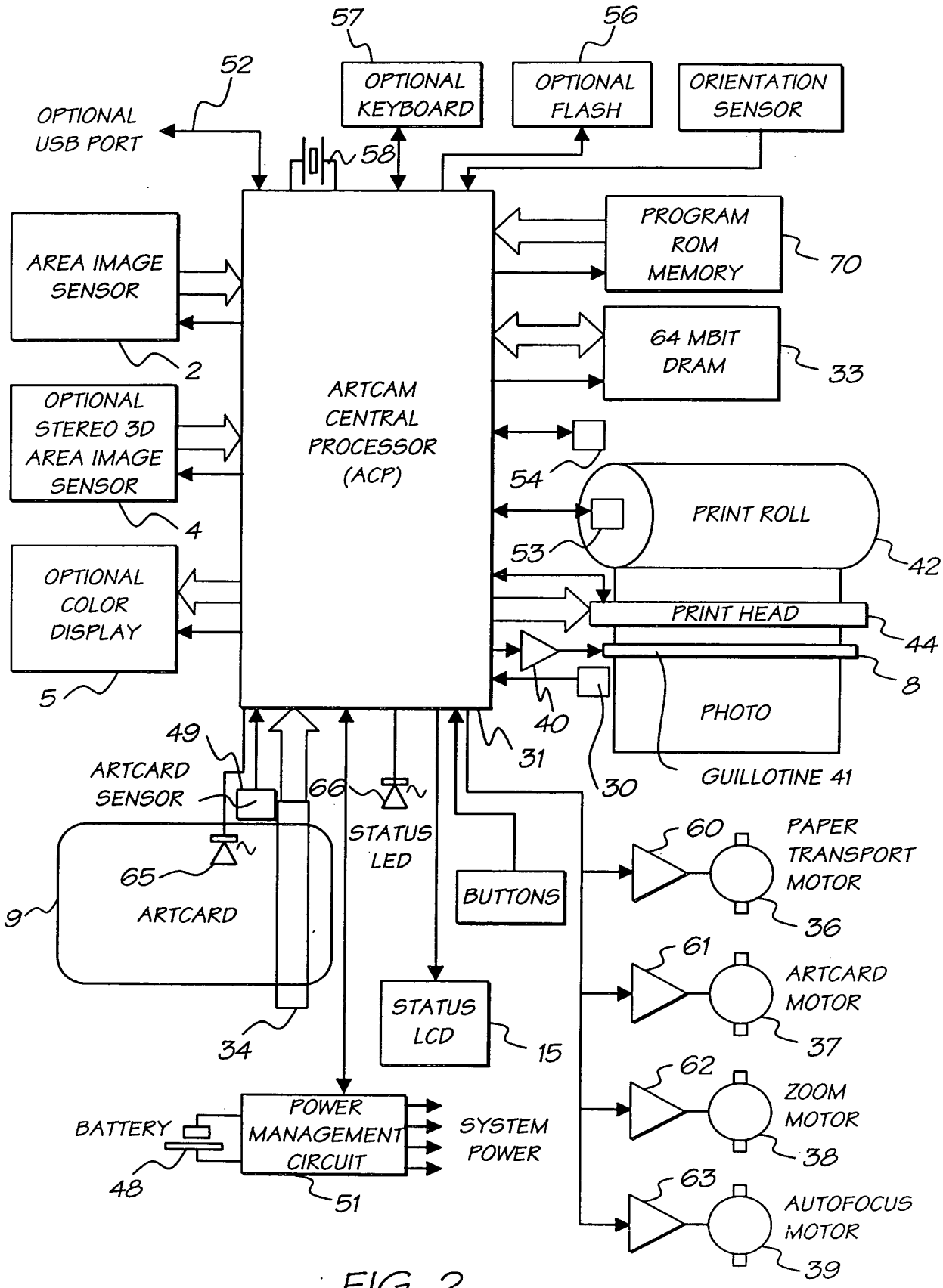


FIG. 2

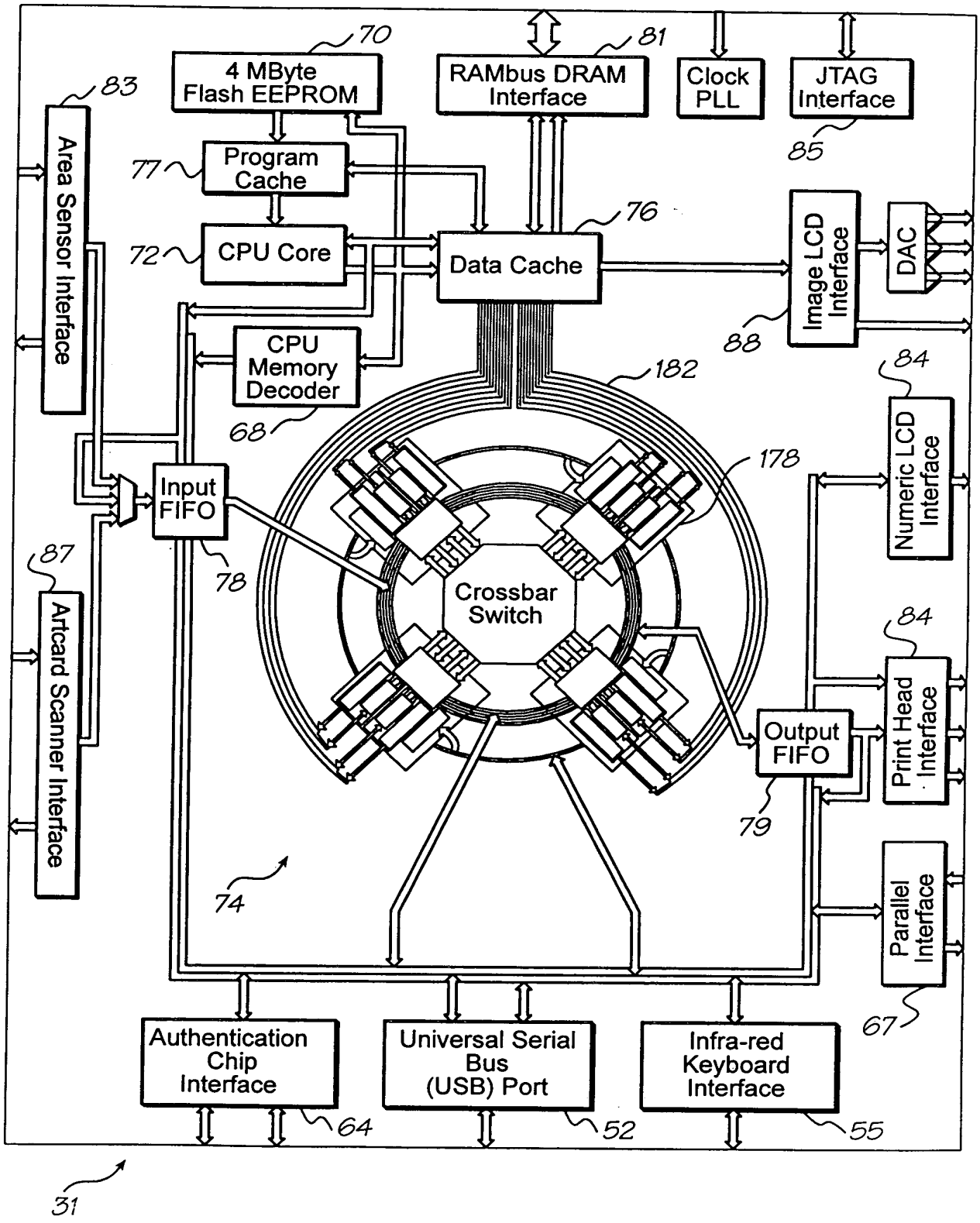

70
4 MByte Flash EEPROM
81
RAMbus DRAM Interface
Clock PLL
JTAG Interface
85
83
Area Sensor Interface
77
Program Cache
72
CPU Core
76
Data Cache
Image LCD Interface
DAC
88
CPU Memory Decoder
68
182
84
Numeric LCD Interface
178
Input FIFO
78
87
Artcard Scanner Interface
Crossbar Switch
84
Print Head Interface
Output FIFO
79
74
Parallel Interface
67
Authentication Chip Interface
64
Universal Serial Bus (USB) Port
52
Infra-red Keyboard Interface
55
31

FIG. 3

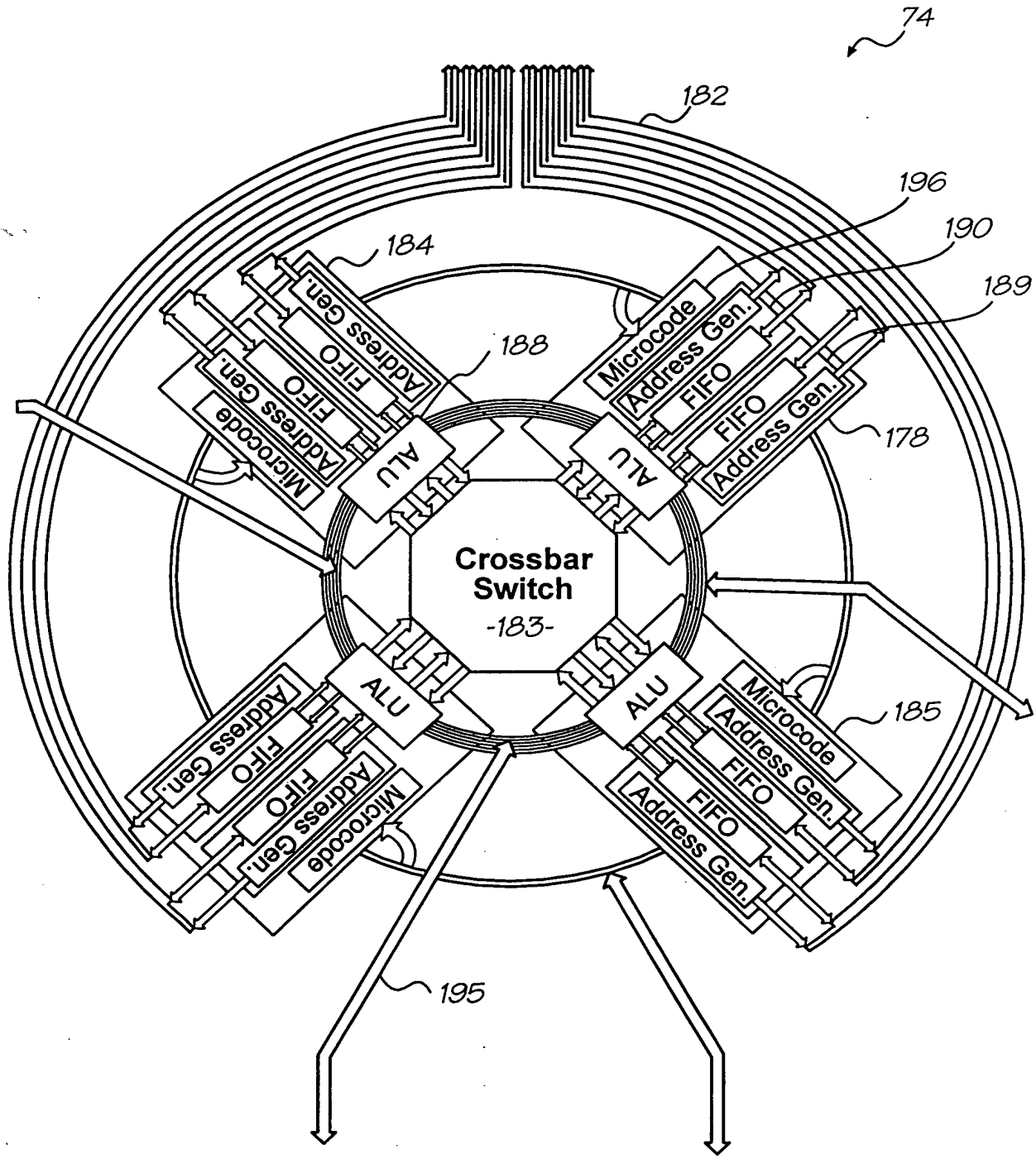

74
182
196
190
189
184
188
178
185
195
Crossbar Switch
-183-
ALU
Microcode
Address Gen.
FIFO
Address Gen.
FIFO
Microcode
Address Gen.
FIFO
Address Gen.
FIFO
Microcode
Address Gen.
FIFO
Address Gen.
FIFO
Microcode
Address Gen.
FIFO
Address Gen.

FIG. 3(a)

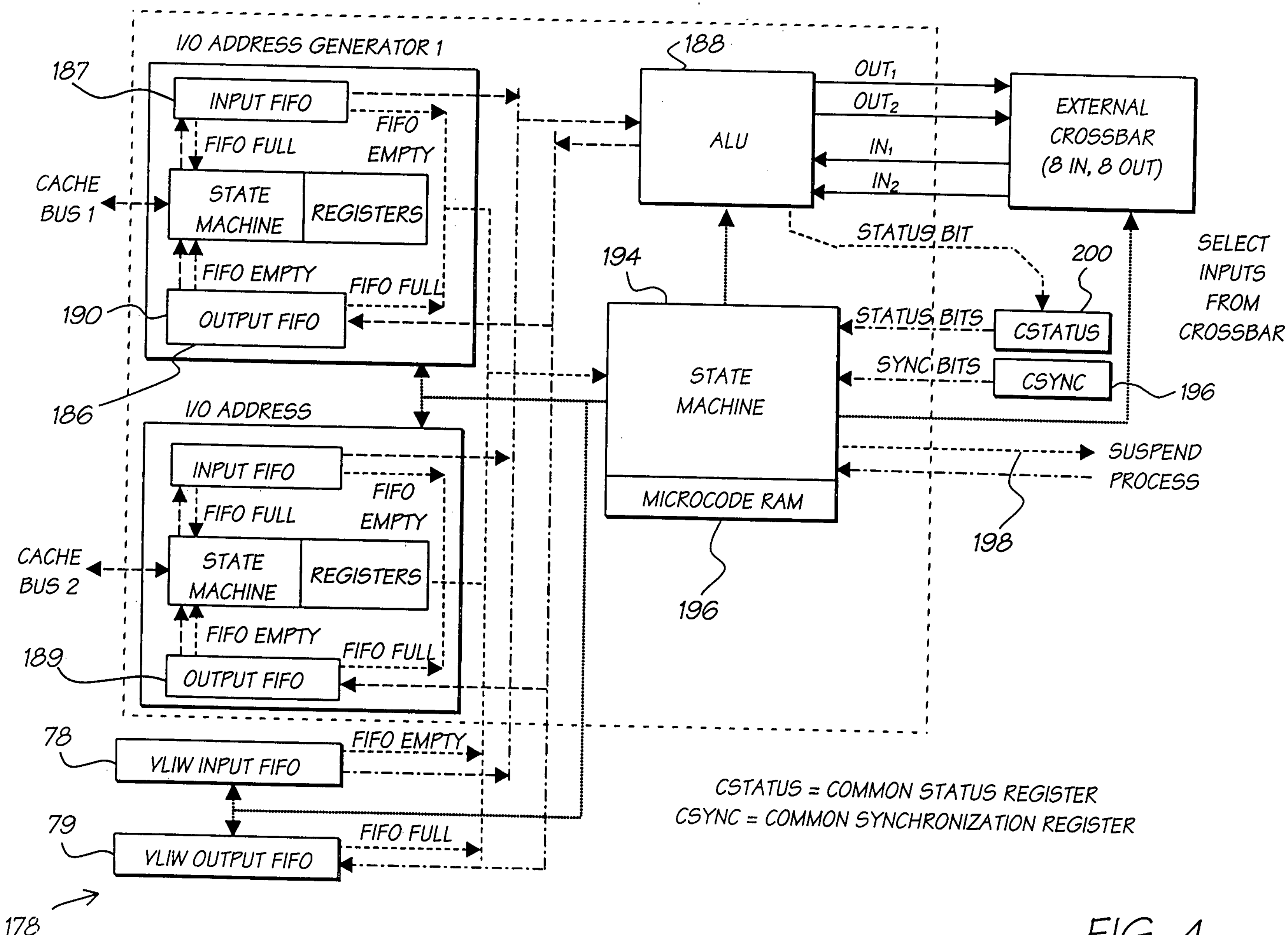

I/O ADDRESS GENERATOR 1
INPUT FIFO
FIFO FULL
FIFO EMPTY
STATE MACHINE
REGISTERS
FIFO EMPTY
FIFO FULL
OUTPUT FIFO
CACHE BUS 1
187
190
186
I/O ADDRESS
INPUT FIFO
FIFO FULL
FIFO EMPTY
STATE MACHINE
REGISTERS
FIFO EMPTY
FIFO FULL
OUTPUT FIFO
CACHE BUS 2
189
VLIW INPUT FIFO
FIFO EMPTY
VLIW OUTPUT FIFO
FIFO FULL
78
79
178
188
ALU
OUT1
OUT2
IN1
IN2
EXTERNAL CROSSBAR (8 IN, 8 OUT)
SELECT INPUTS FROM CROSSBAR
STATUS BIT
194
STATE MACHINE
MICROCODE RAM
196
200
STATUS BITS
CSTATUS
SYNC BITS
CSYNC
196
SUSPEND PROCESS
198
CSTATUS = COMMON STATUS REGISTER
CSYNC = COMMON SYNCHRONIZATION REGISTER

FIG. 4

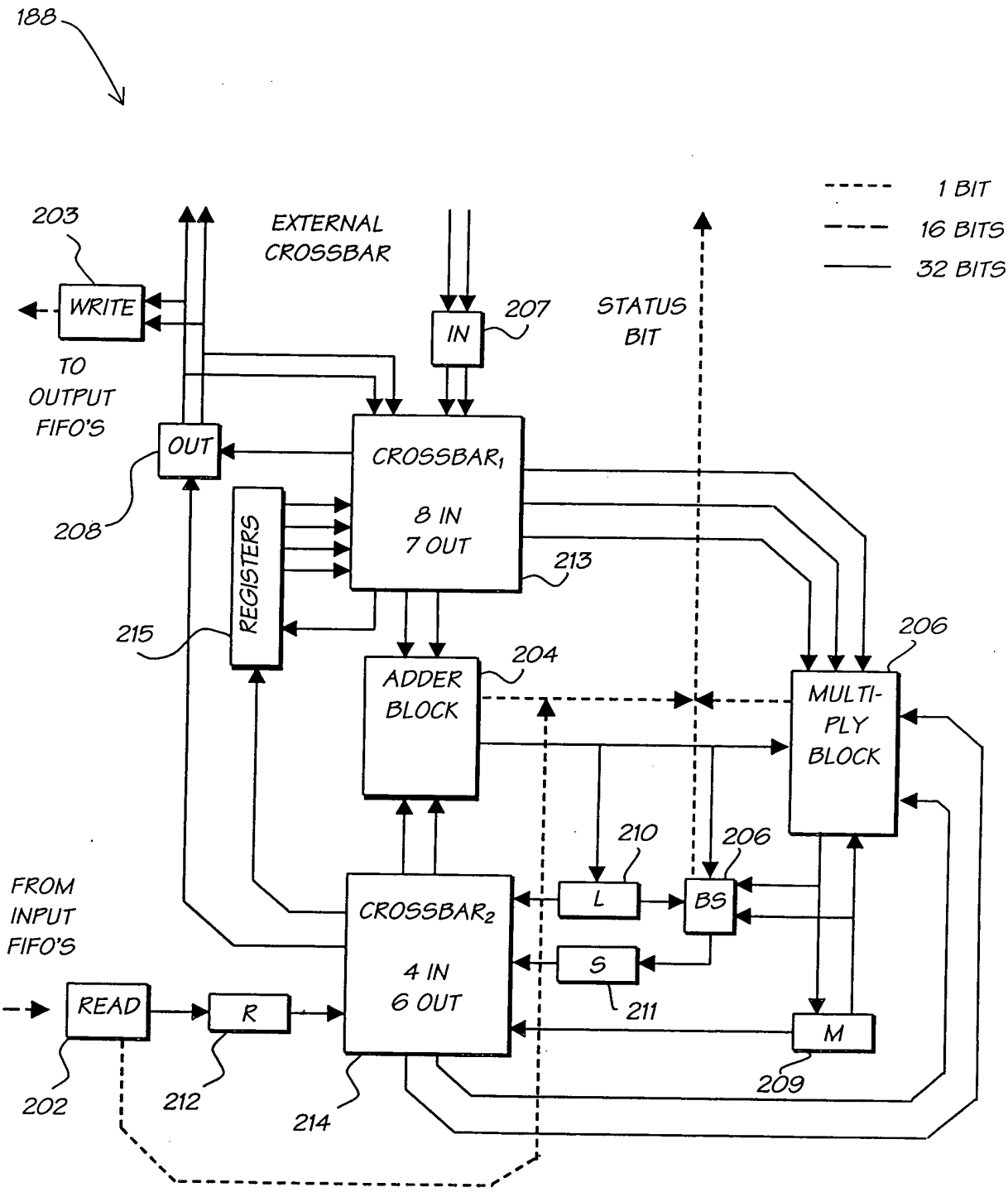

188
1 BIT
16 BITS
32 BITS
203
WRITE
TO OUTPUT FIFO'S
EXTERNAL CROSSBAR
IN
207
STATUS BIT
OUT
208
CROSSBAR1
8 IN 7 OUT
213
REGISTERS
215
ADDER BLOCK
204
MULTI-PLY BLOCK
206
FROM INPUT FIFO'S
CROSSBAR2
4 IN 6 OUT
L
210
BS
206
S
211
READ
R
M
202
212
214
209

FIG. 5

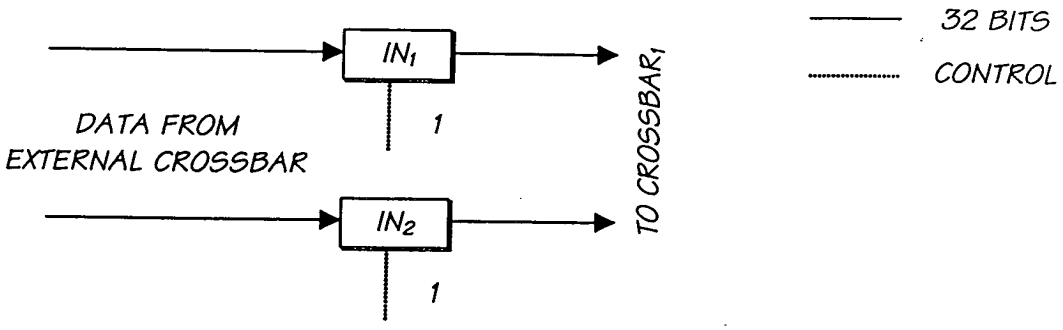

32 BITS
CONTROL
IN1
1
DATA FROM
EXTERNAL CROSSBAR
IN2
1
TO CROSSBAR1

FIG. 6

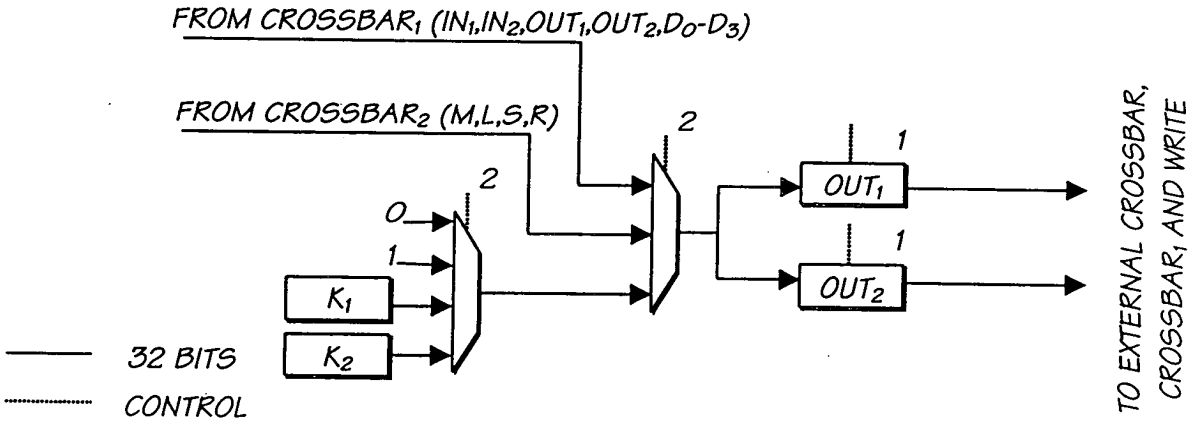

FROM CROSSBAR1 (IN1,IN2,OUT1,OUT2,D0-D3)
FROM CROSSBAR2 (M,L,S,R)
2
2
0
1
K1
K2
1
OUT1
1
OUT2
32 BITS
CONTROL
TO EXTERNAL CROSSBAR,
CROSSBAR1 AND WRITE

FIG. 7

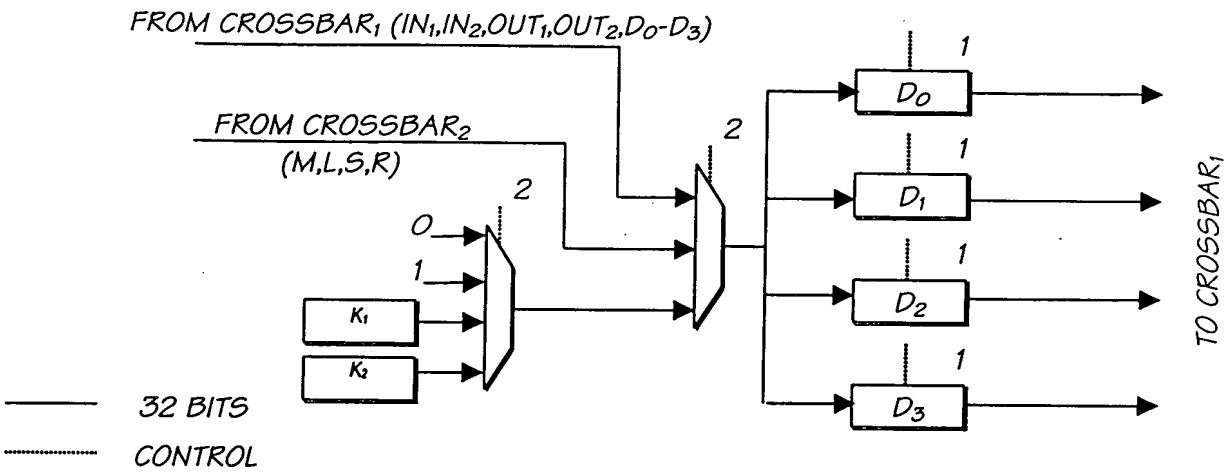

FROM CROSSBAR1 (IN1,IN2,OUT1,OUT2,D0-D3)
FROM CROSSBAR2
(M,L,S,R)
2
2
0
1
K1
K2
1
D0
1
D1
1
D2
1
D3
TO CROSSBAR1
32 BITS
CONTROL

FIG. 8

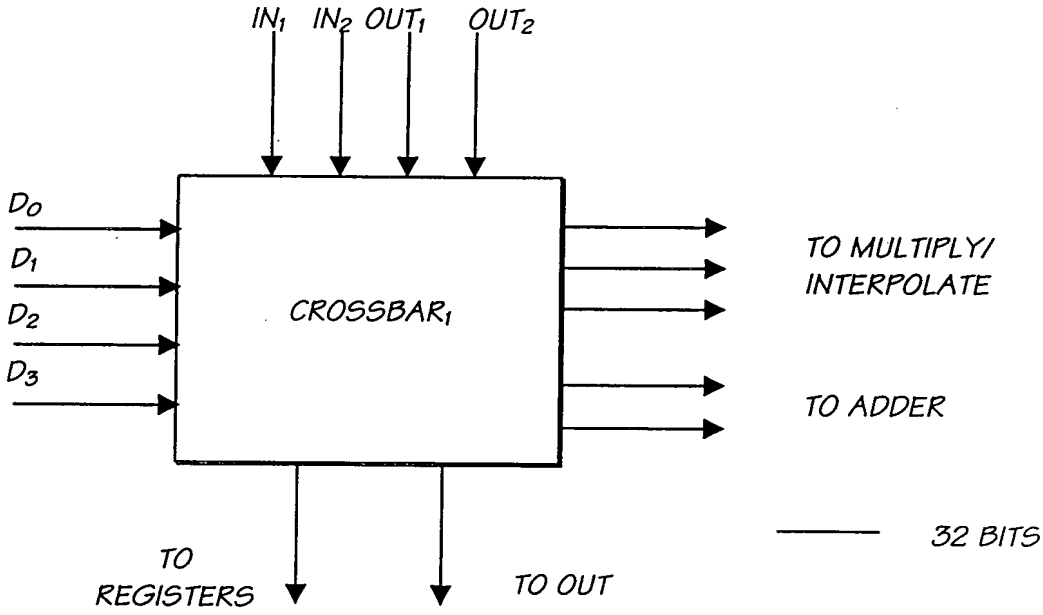

IN1
IN2
OUT1
OUT2
D0
D1
D2
D3
CROSSBAR1
TO MULTIPLY/
INTERPOLATE
TO ADDER
32 BITS
TO
REGISTERS
TO OUT

FIG. 9

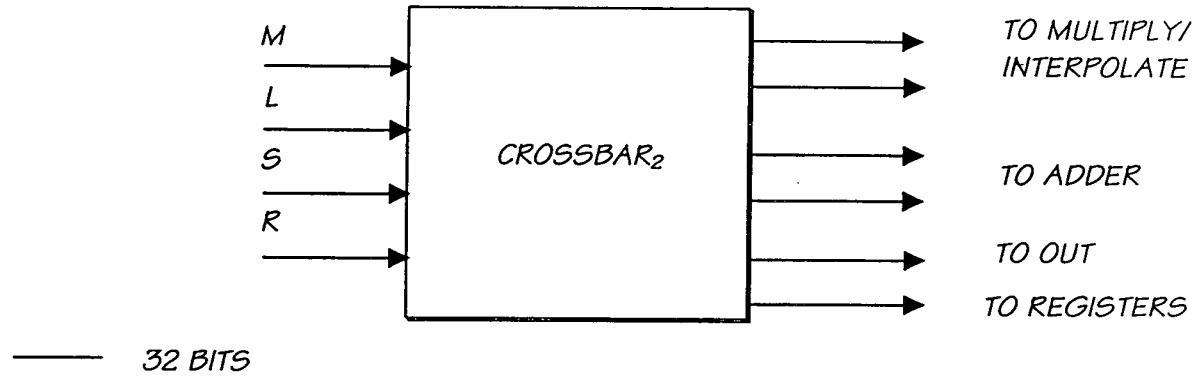

M
L
S
R
CROSSBAR2
TO MULTIPLY/
INTERPOLATE
TO ADDER
TO OUT
TO REGISTERS
32 BITS

FIG. 10

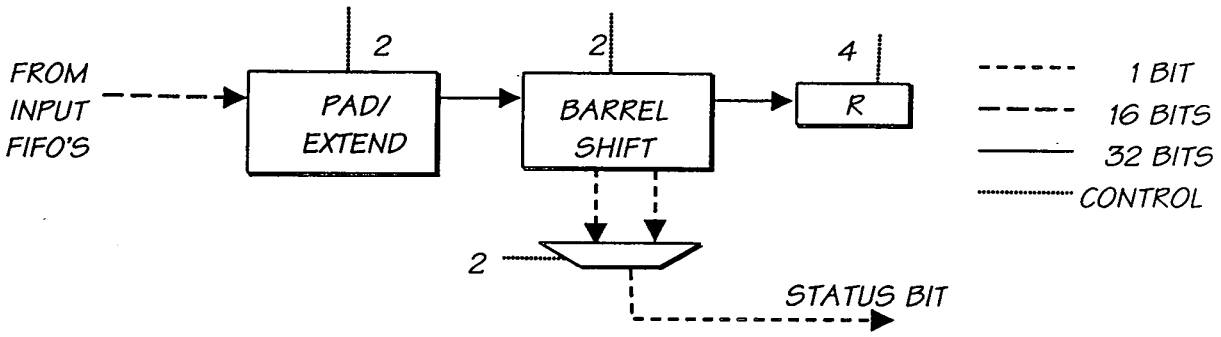

FROM
INPUT
FIFO'S
2
PAD/
EXTEND
2
BARREL
SHIFT
4
R
1 BIT
16 BITS
32 BITS
CONTROL
2
STATUS BIT

FIG. 11

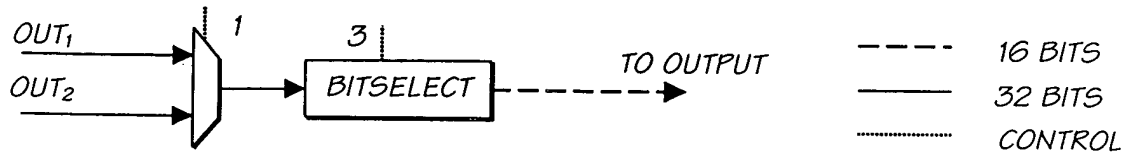


FIG. 12

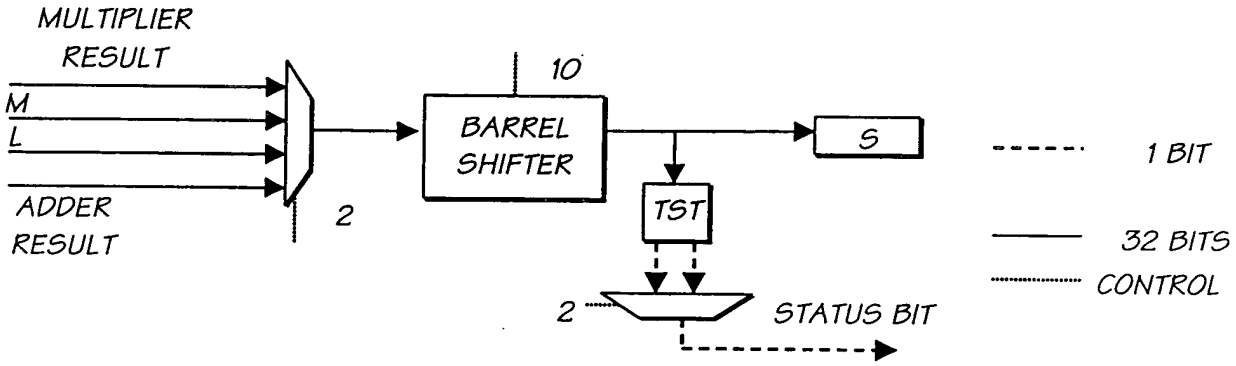


FIG. 13

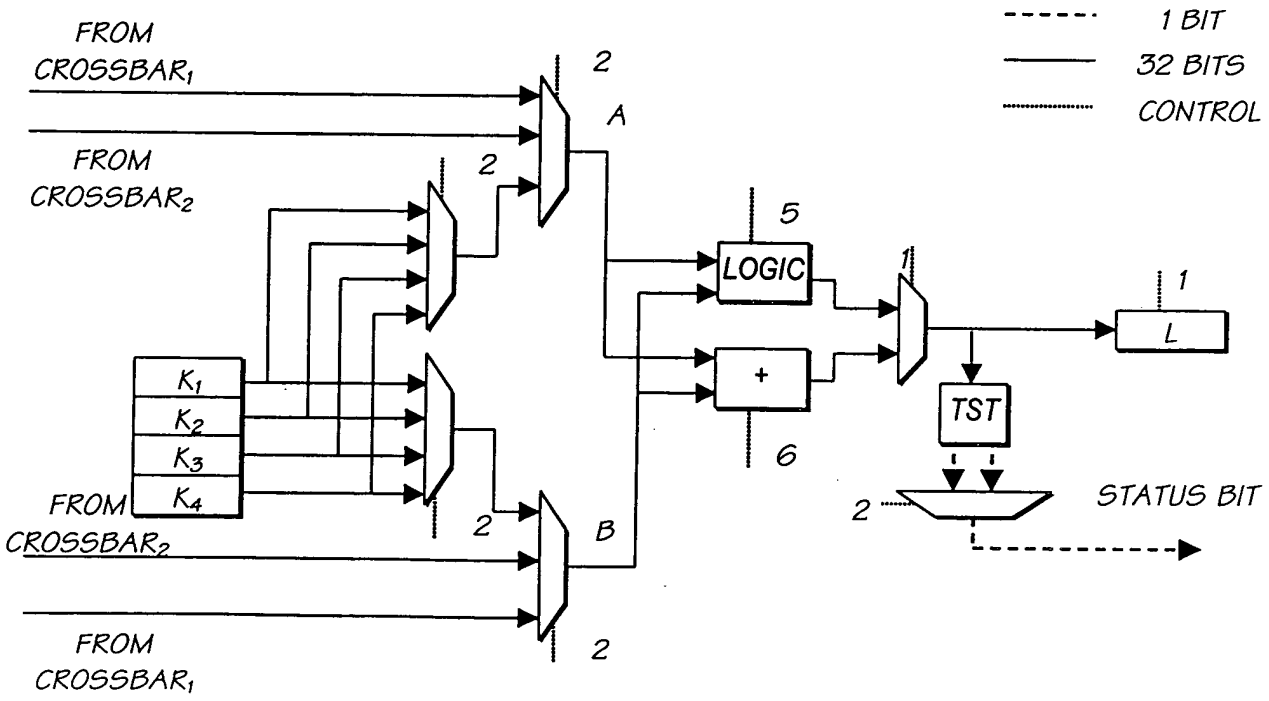


FIG. 14

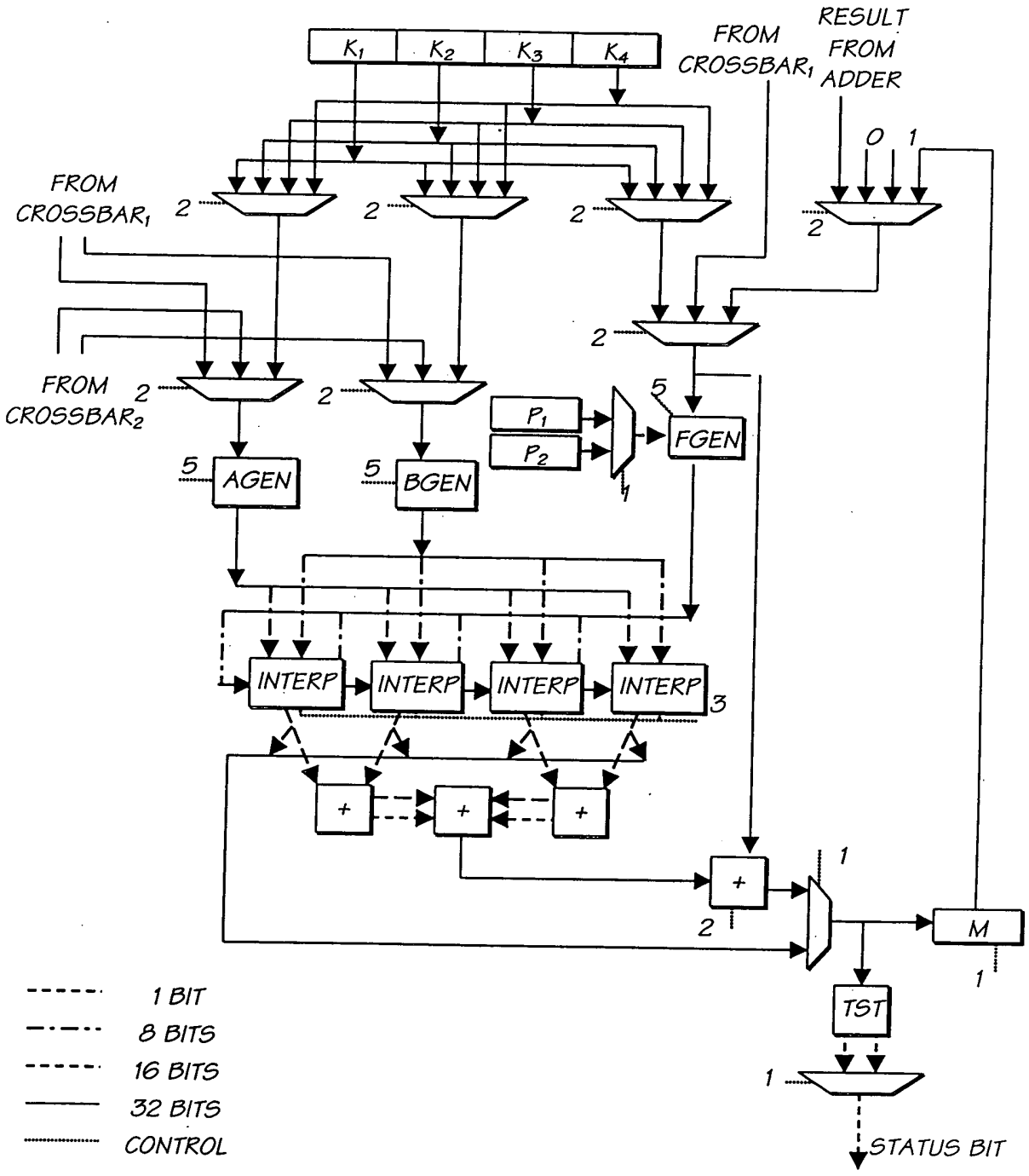

K1
K2
K3
K4
FROM CROSSBAR1
RESULT FROM ADDER
FROM CROSSBAR1
FROM CROSSBAR2
AGEN
BGEN
P1
P2
FGEN
INTERP
INTERP
INTERP
INTERP
M
TST
STATUS BIT
1 BIT
8 BITS
16 BITS
32 BITS
CONTROL

FIG. 15

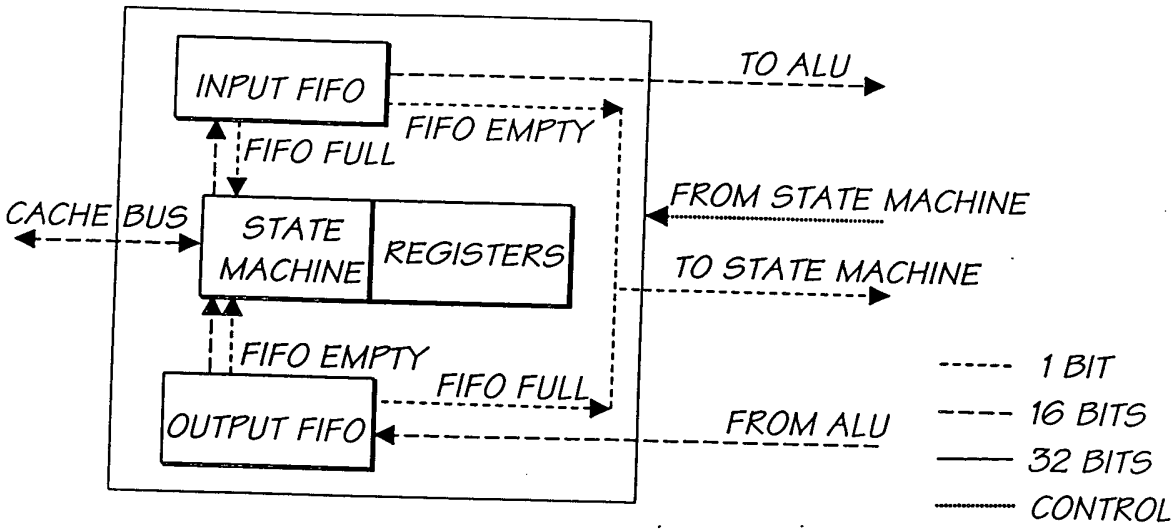


FIG. 16

ORDER OF PIXELS PRESENTED BY A SEQUENTIAL READ ITERATOR ON A 4 X 2 IMAGE WITH PADDING.

0	1	2	3	
4	5	6	7	

FIG. 17

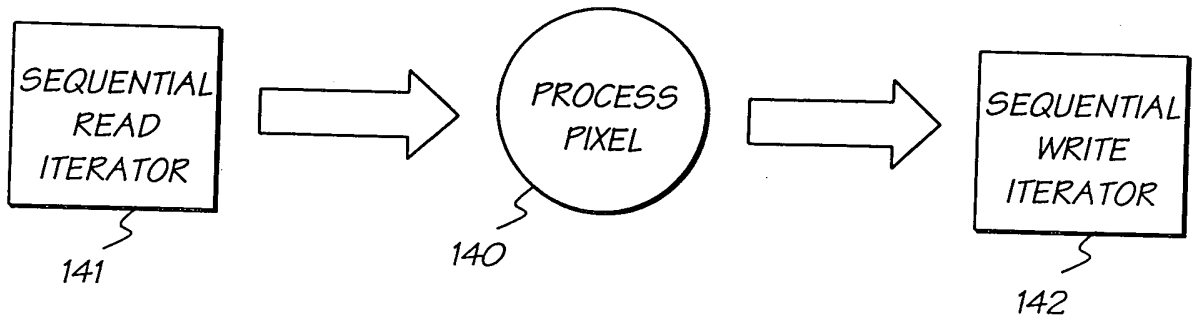
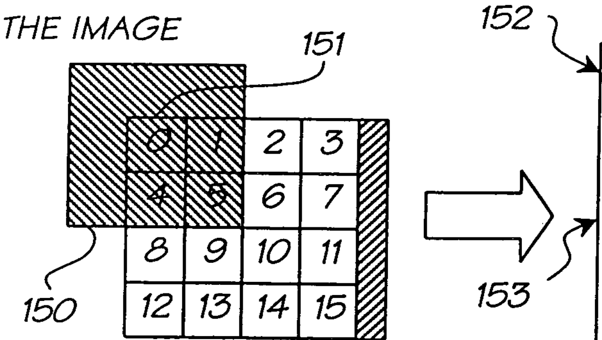


FIG. 18

A 3x3 BOX VIEW TRAVERSES THE PIXELS IN ORDER: 0, 1, 2, 3, 4, 5, 6, 7, 8 ETC, PLACING A 3x3 BOX CENTERED OVER EACH PIXEL...

3x3 BOX VIEW OF FIRST PIXEL IN IMAGE = 9 PIXELS, 5 OF WHICH ARE OUTSIDE THE IMAGE

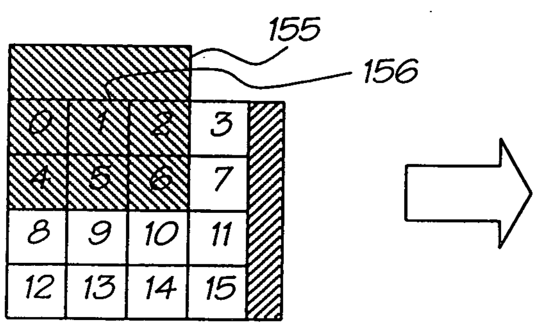


FIRST 9 PIXELS FROM THE BOX READ ITERATOR:

IF DUPLICATION OF EDGE PIXELS IS ON: 0, 0, 0, 0, 0, 1, 4, 4, 5

IF DUPLICATION OF EDGE PIXELS IS OFF: V, V, V, V, 0, 1, V, 4, 5 WHERE V IS CONSTANTPIXEL REGISTER VALUE REPRESENTING "OUTSIDE THE IMAGE"

3x3 BOX VIEW OF SECOND PIXEL IN IMAGE = 9 PIXELS, 3 OF WHICH ARE OUTSIDE THE IMAGE



SECOND 9 PIXELS FROM THE BOX READ ITERATOR:

IF DUPLICATION OF EDGE PIXELS IS ON: 0, 1, 2, 0, 1, 2, 4, 5, 6

IF DUPLICATION OF EDGE PIXELS IS OFF: V, V, V, 0, 1, 2, 4, 5, 6 WHERE V IS CONSTANTPIXEL REGISTER VALUE REPRESENTING "OUTSIDE THE IMAGE"

FIG. 19

10522275.080504

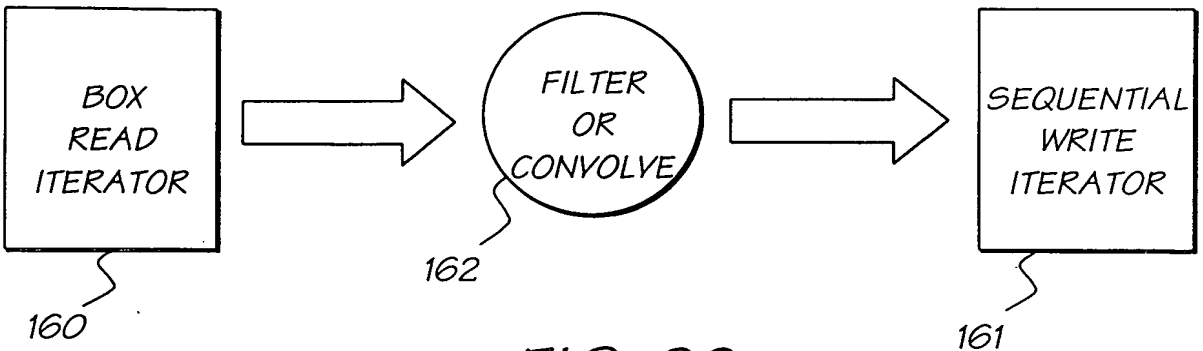


FIG. 20

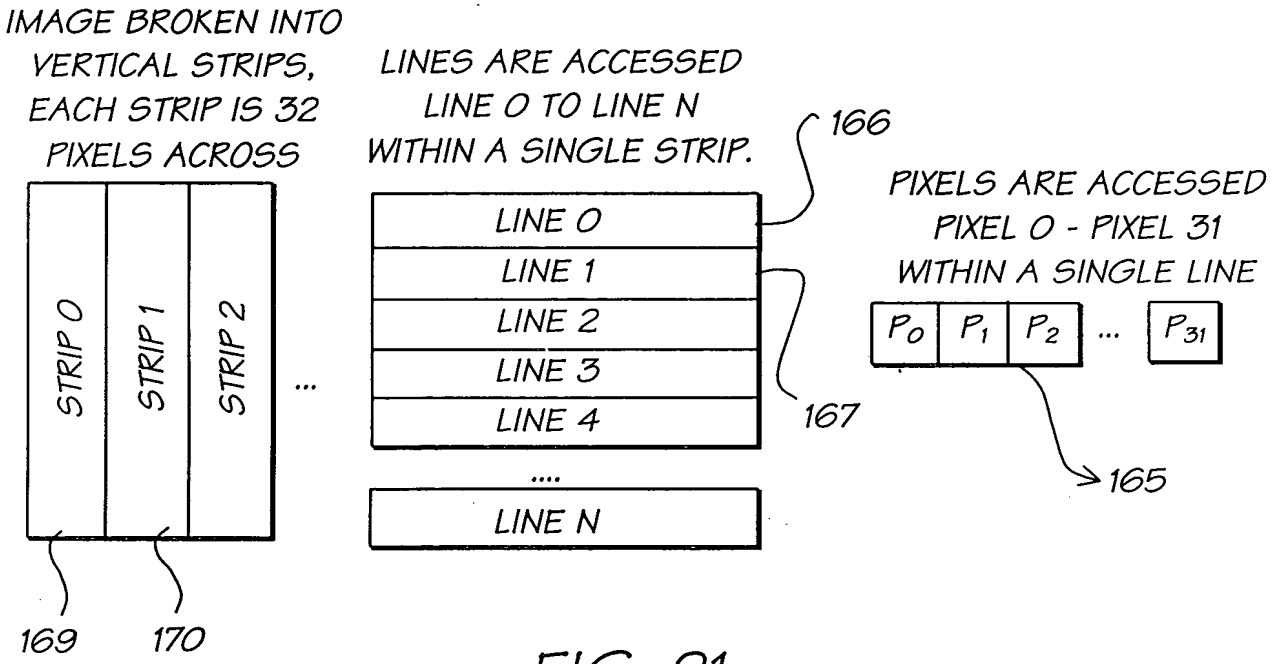


FIG. 21

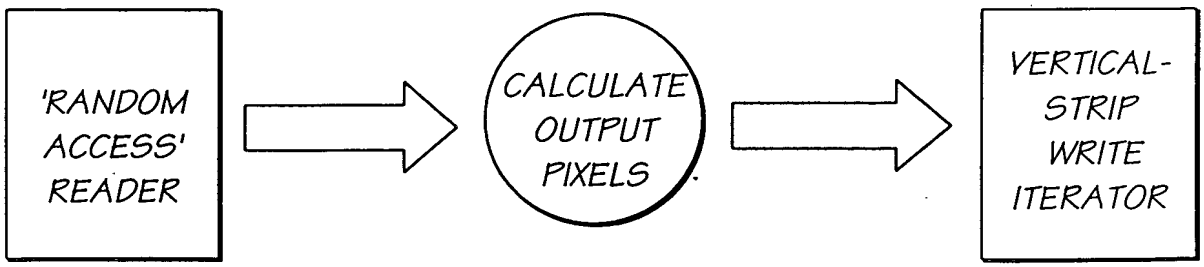


FIG. 22

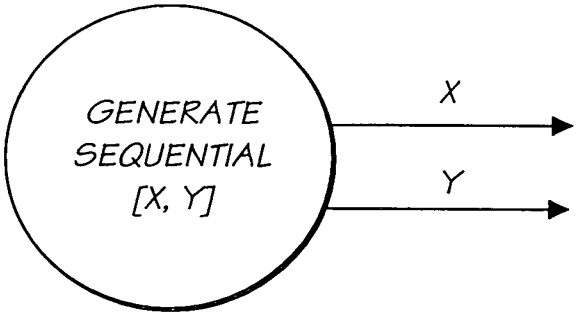

GENERATE
SEQUENTIAL
[X, Y]
X
Y

FIG. 23

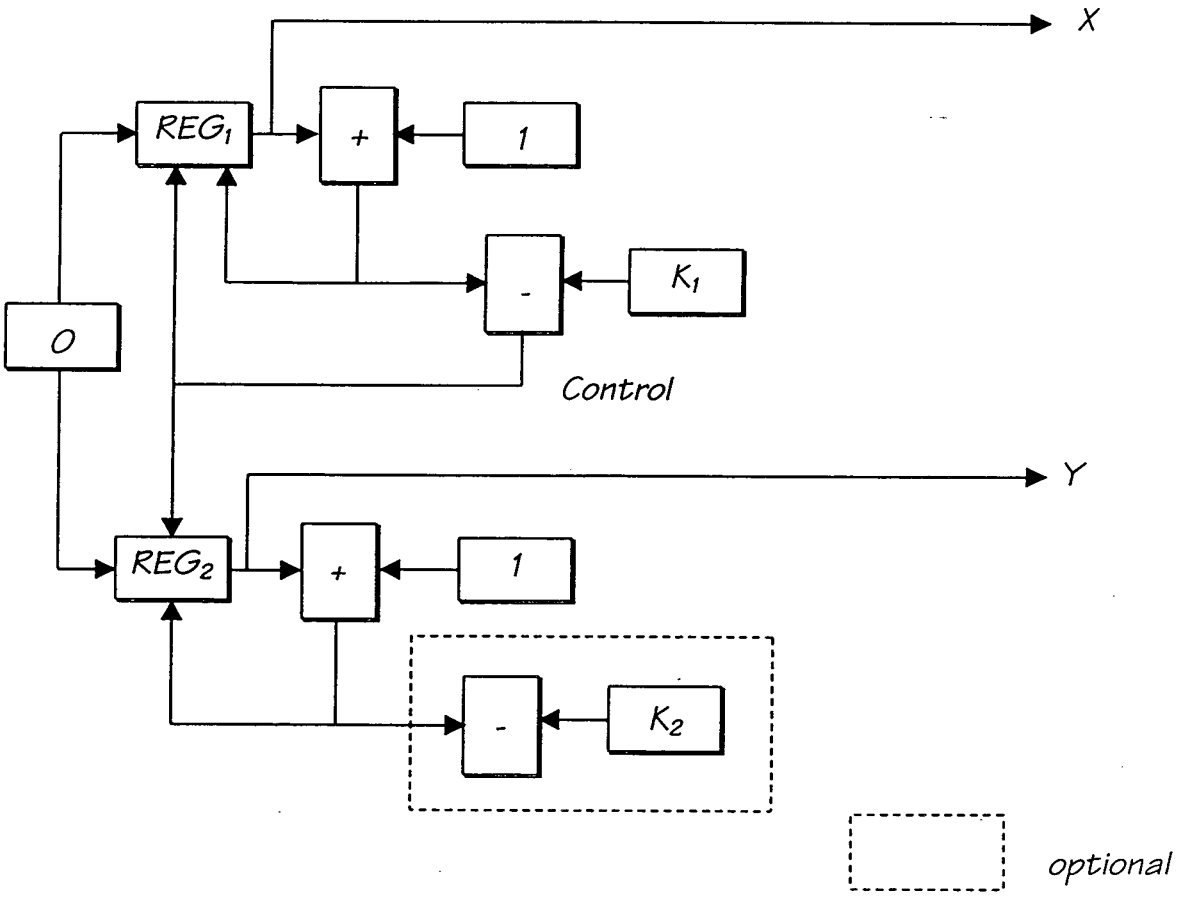

X
REG1
+
1
-
K1
0
Control
Y
REG2
+
1
-
K2
optional

FIG. 24

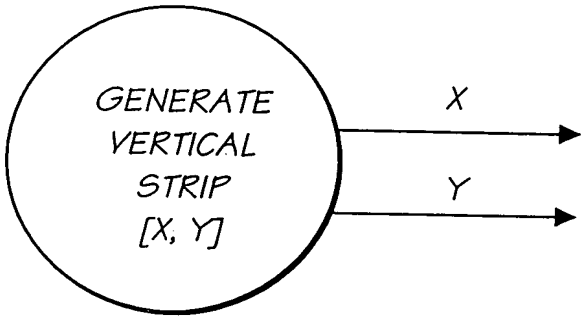

GENERATE
VERTICAL
STRIP
[X, Y]
X
Y

FIG. 25

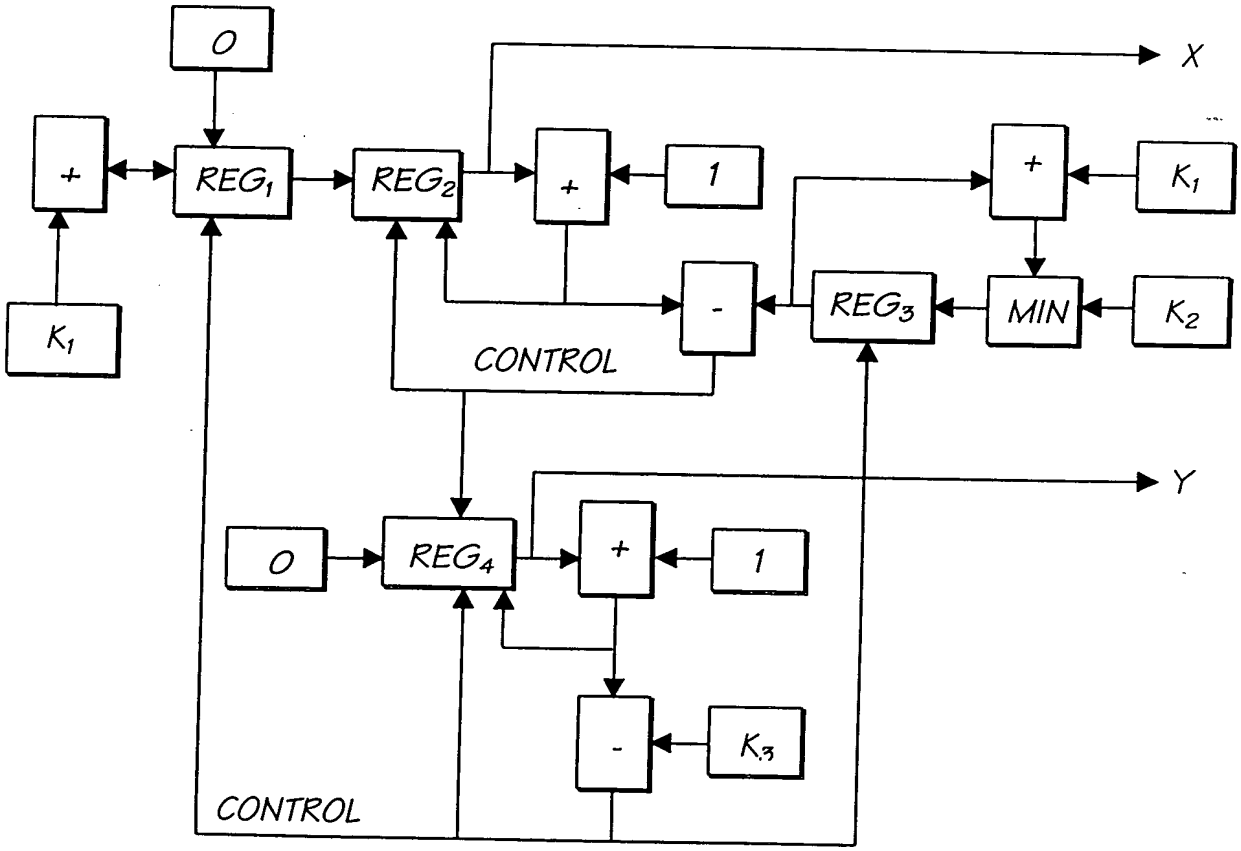
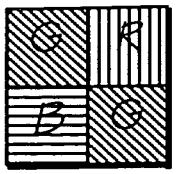

0
X
+
REG1
REG2
+
1
+
K1
K1
-
REG3
MIN
K2
CONTROL
Y
0
REG4
+
1
-
K3
CONTROL

FIG. 26



2X2 PIXEL BLOCK FROM SENSOR

FIG. 27

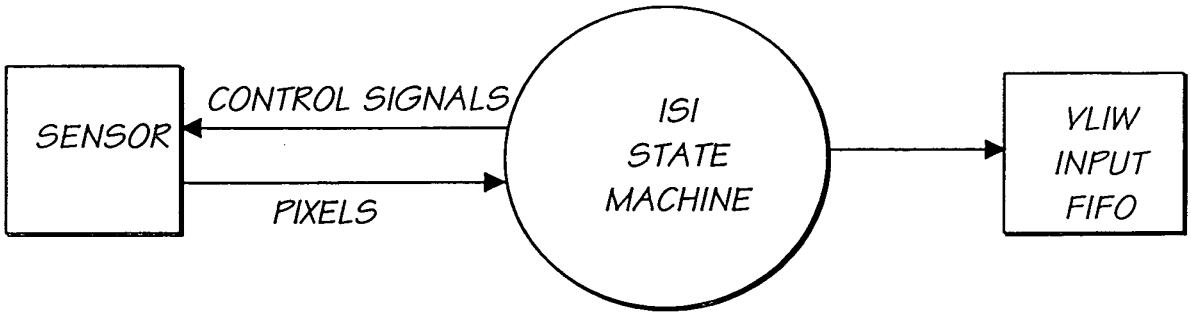


FIG. 28

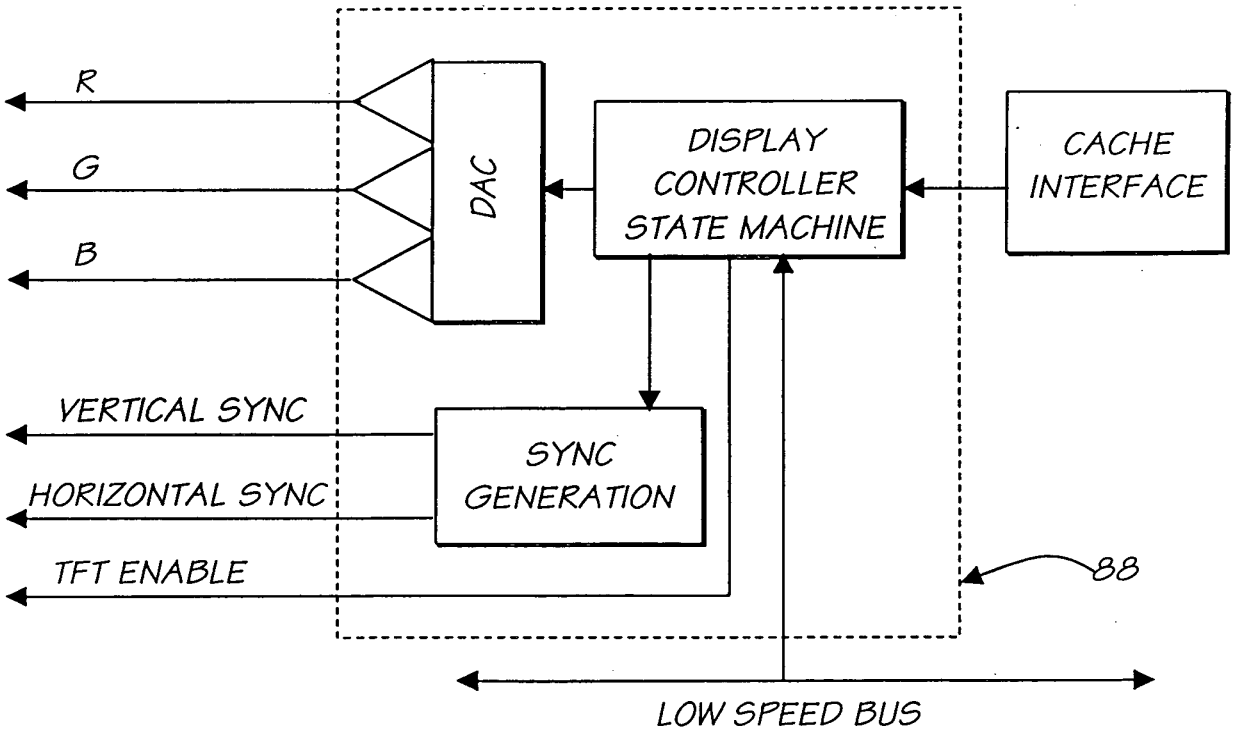
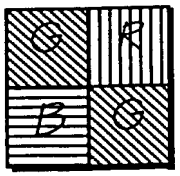


FIG. 29



2X2 PIXEL BLOCK FROM CCD

FIG. 30

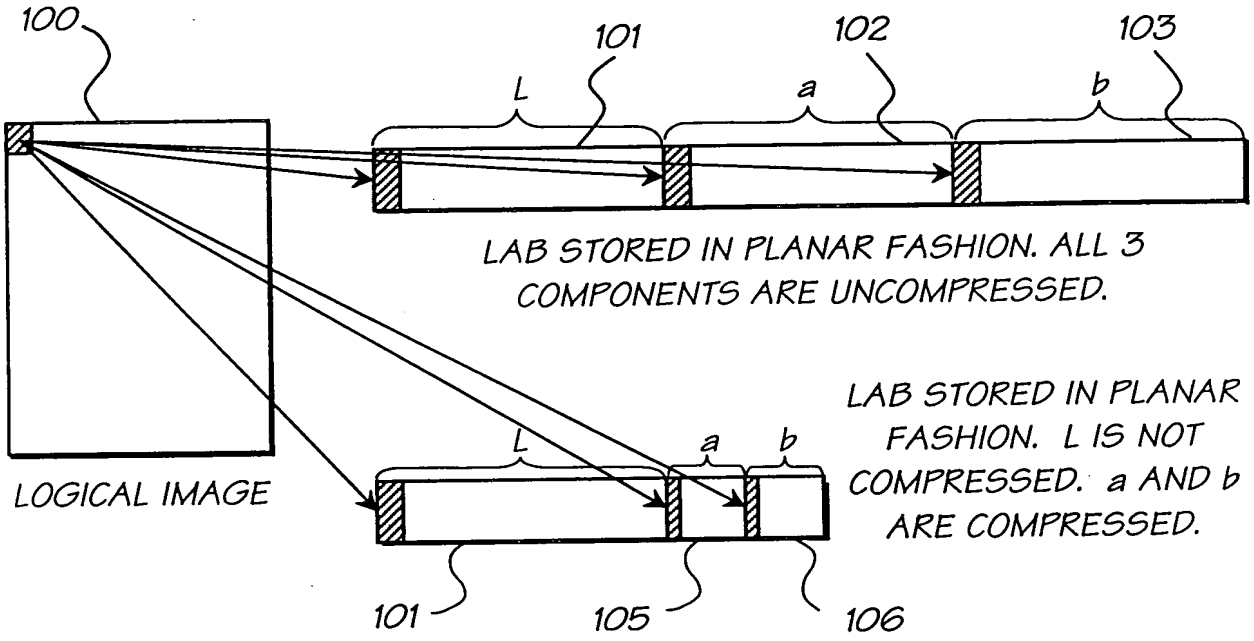


FIG. 31

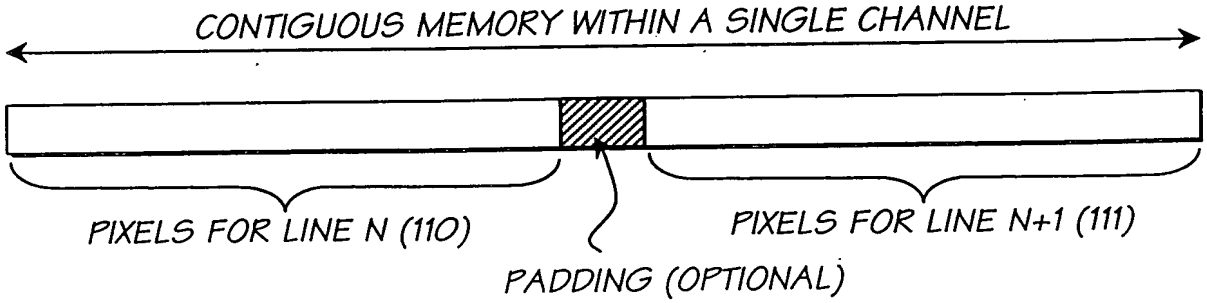


FIG. 32

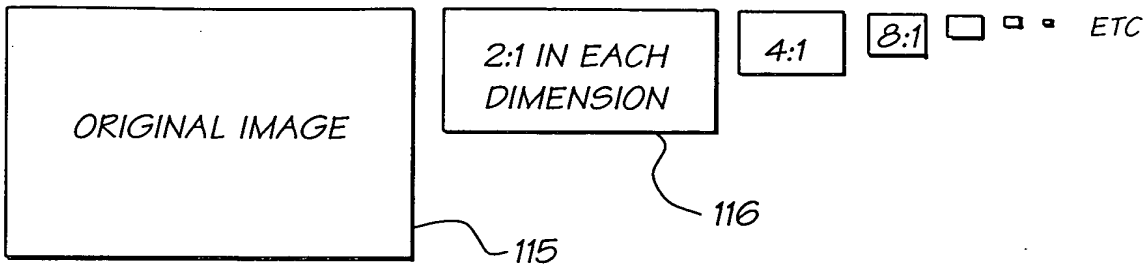


FIG. 33

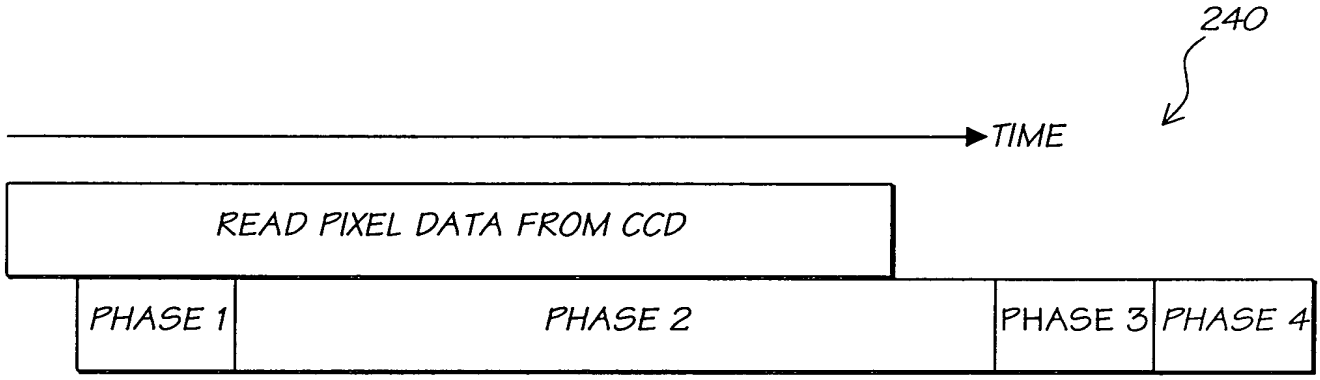


FIG. 34

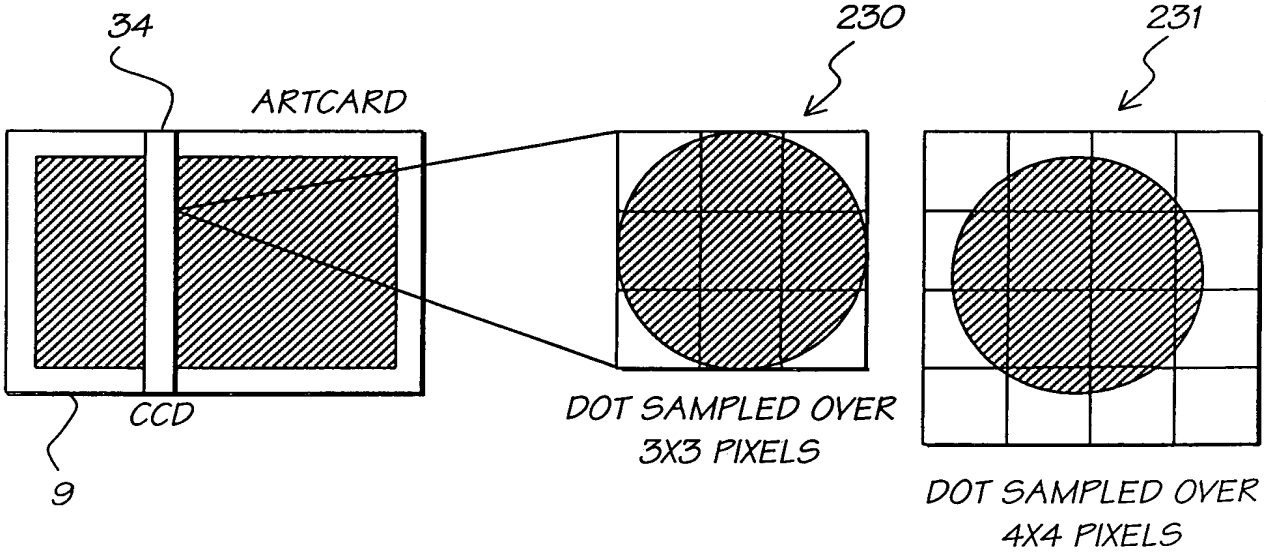


FIG. 35

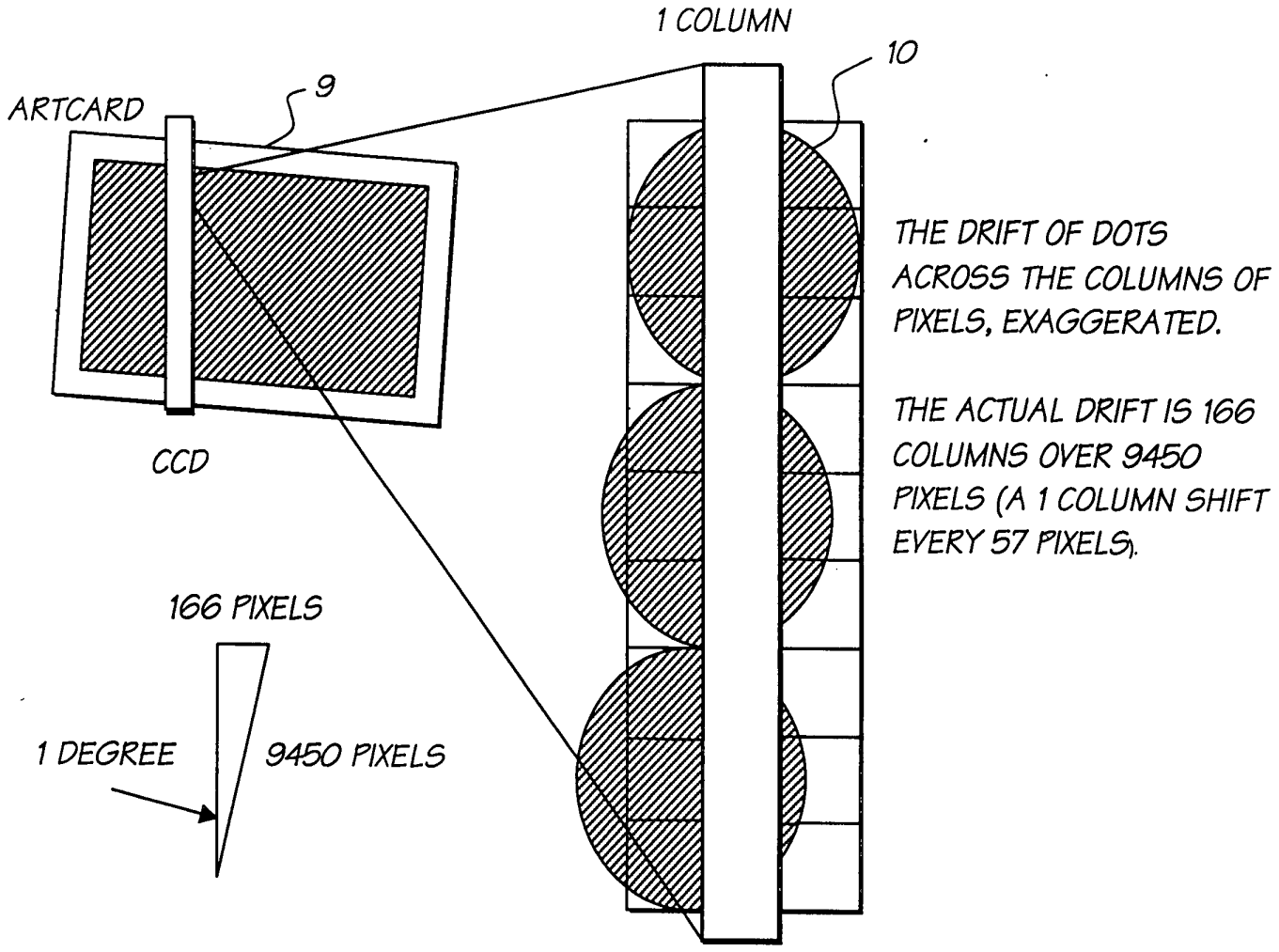

1 COLUMN
10
ARTCARD
9
CCD
166 PIXELS
1 DEGREE
9450 PIXELS
THE DRIFT OF DOTS ACROSS THE COLUMNS OF PIXELS, EXAGGERATED.
THE ACTUAL DRIFT IS 166 COLUMNS OVER 9450 PIXELS (A 1 COLUMN SHIFT EVERY 57 PIXELS).

FIG. 36

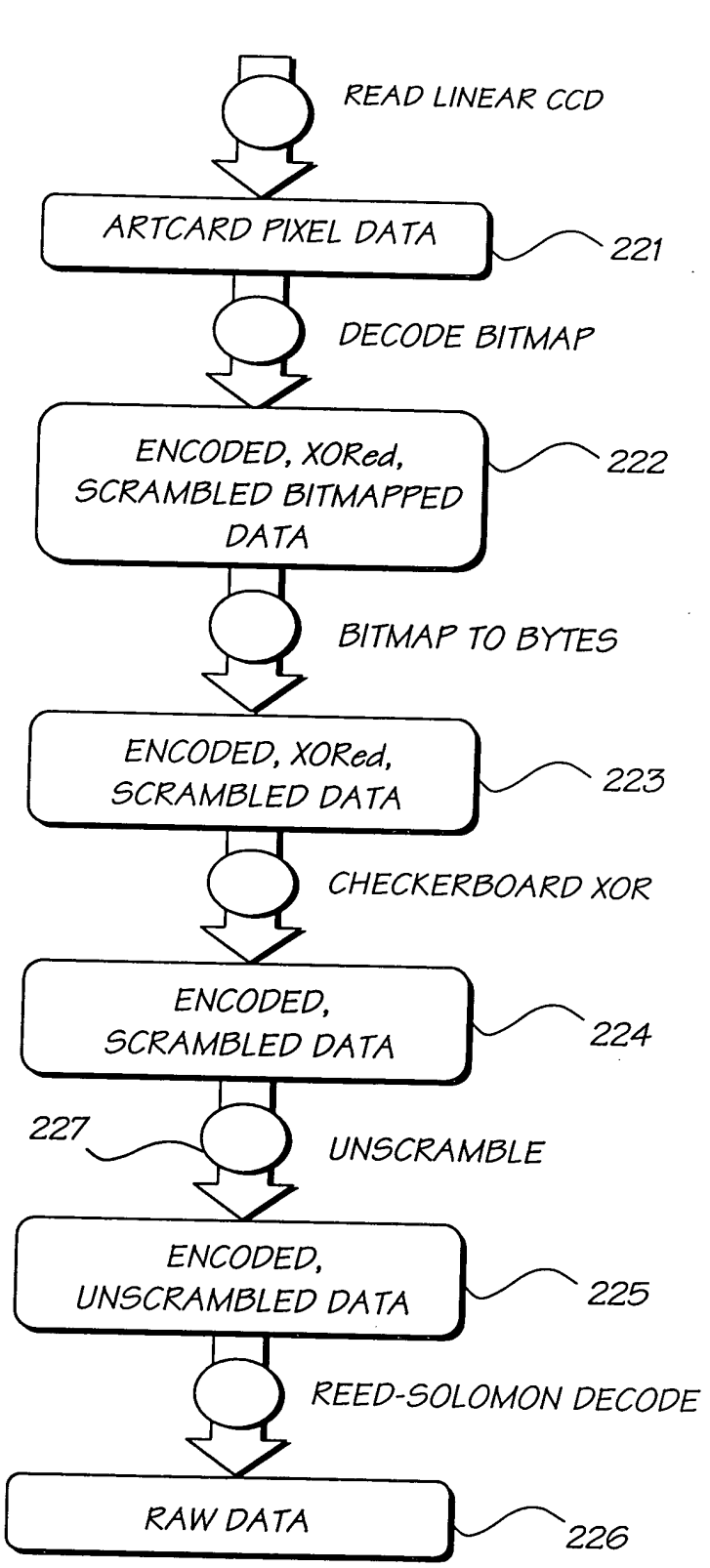

220
READ LINEAR CCD
ARTCARD PIXEL DATA
221
DECODE BITMAP
ENCODED, XORed, SCRAMBLED BITMAPPED DATA
222
BITMAP TO BYTES
ENCODED, XORed, SCRAMBLED DATA
223
CHECKERBOARD XOR
ENCODED, SCRAMBLED DATA
224
227
UNSCRAMBLE
ENCODED, UNSCRAMBLED DATA
225
REED-SOLOMON DECODE
RAW DATA
226

FIG. 37

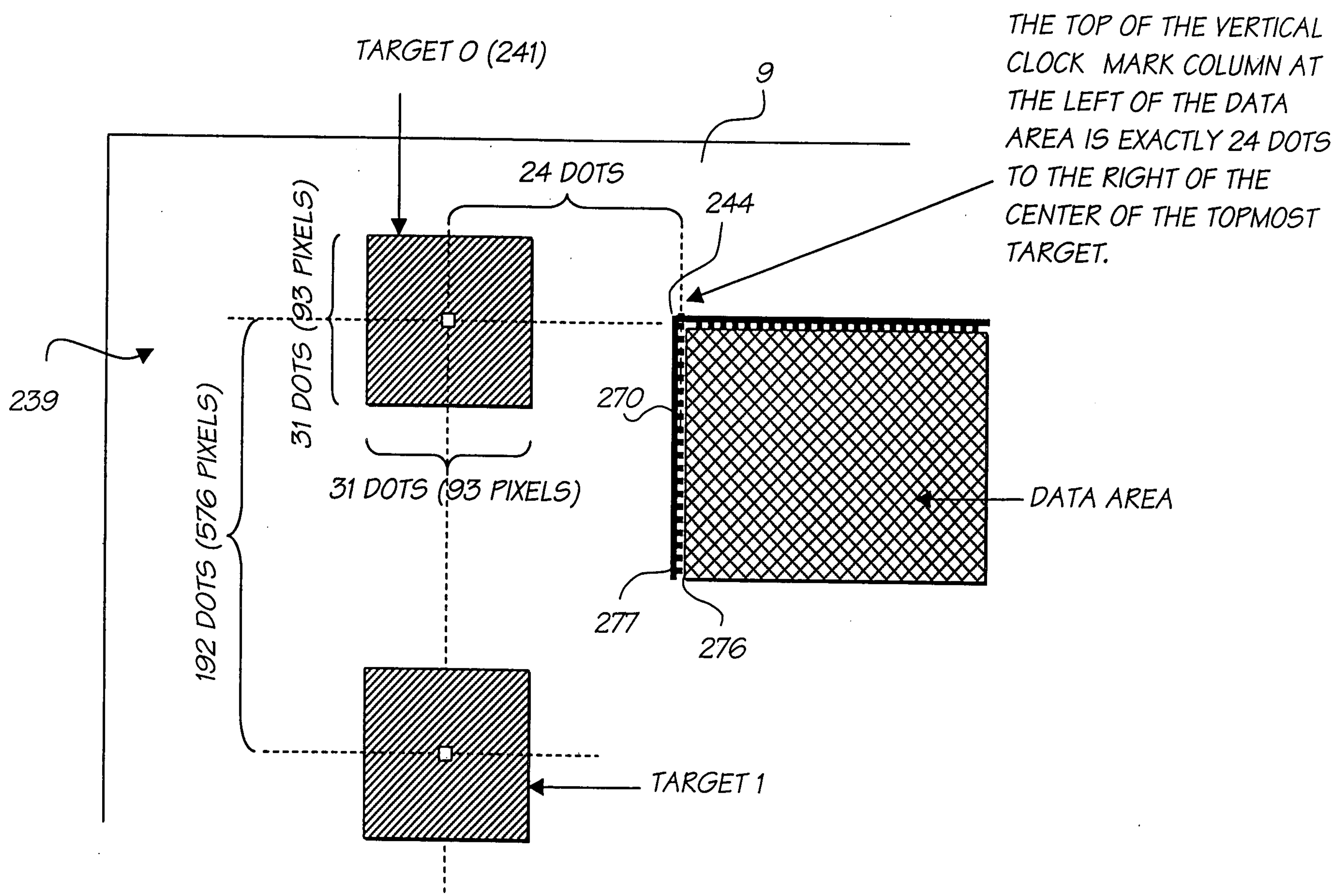

THE TOP OF THE VERTICAL CLOCK MARK COLUMN AT THE LEFT OF THE DATA AREA IS EXACTLY 24 DOTS TO THE RIGHT OF THE CENTER OF THE TOPMOST TARGET.
TARGET 0 (241)
9
24 DOTS
244
239
31 DOTS (93 PIXELS)
31 DOTS (93 PIXELS)
270
DATA AREA
192 DOTS (576 PIXELS)
277
276
TARGET 1

FIG. 38

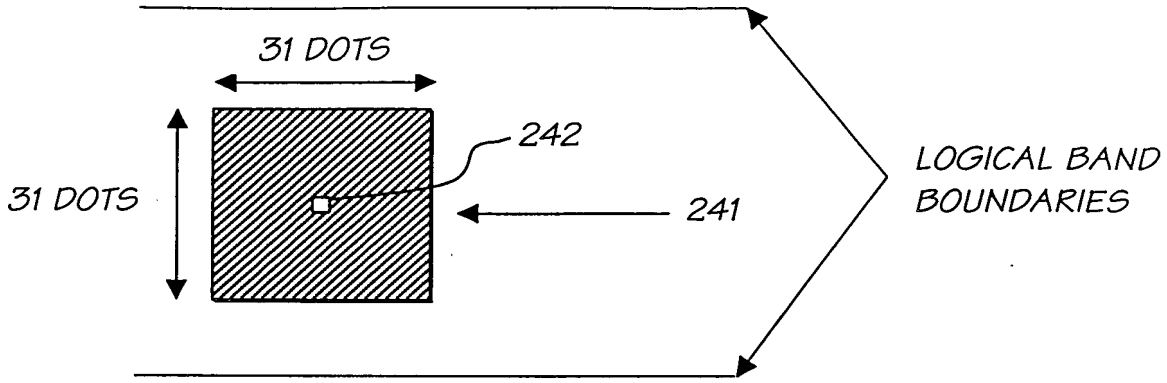

31 DOTS
31 DOTS
242
241
LOGICAL BAND
BOUNDARIES

FIG. 39

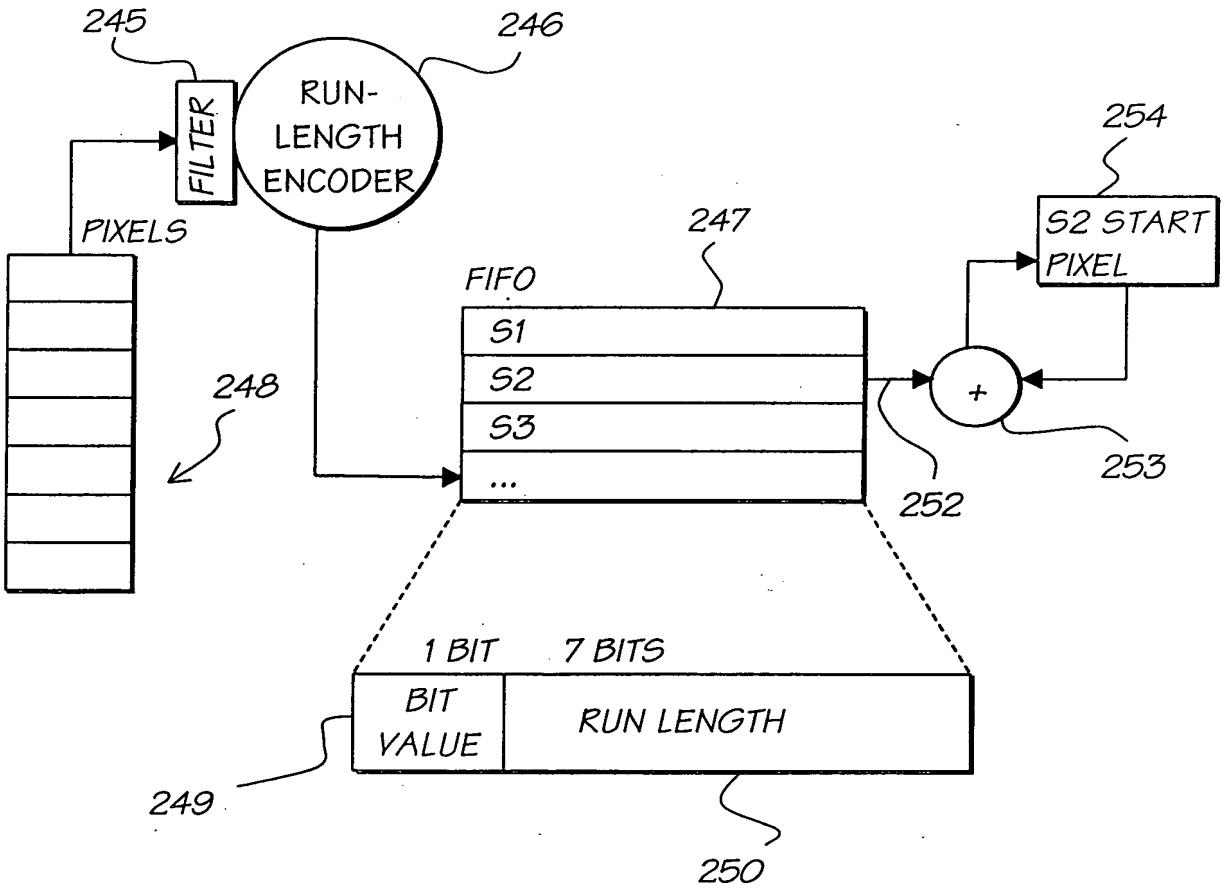

245
246
RUN-
LENGTH
ENCODER
FILTER
PIXELS
247
FIFO
S1
S2
S3
...
254
S2 START
PIXEL
+
253
252
248
1 BIT
7 BITS
BIT
VALUE
RUN LENGTH
249
250

FIG. 40

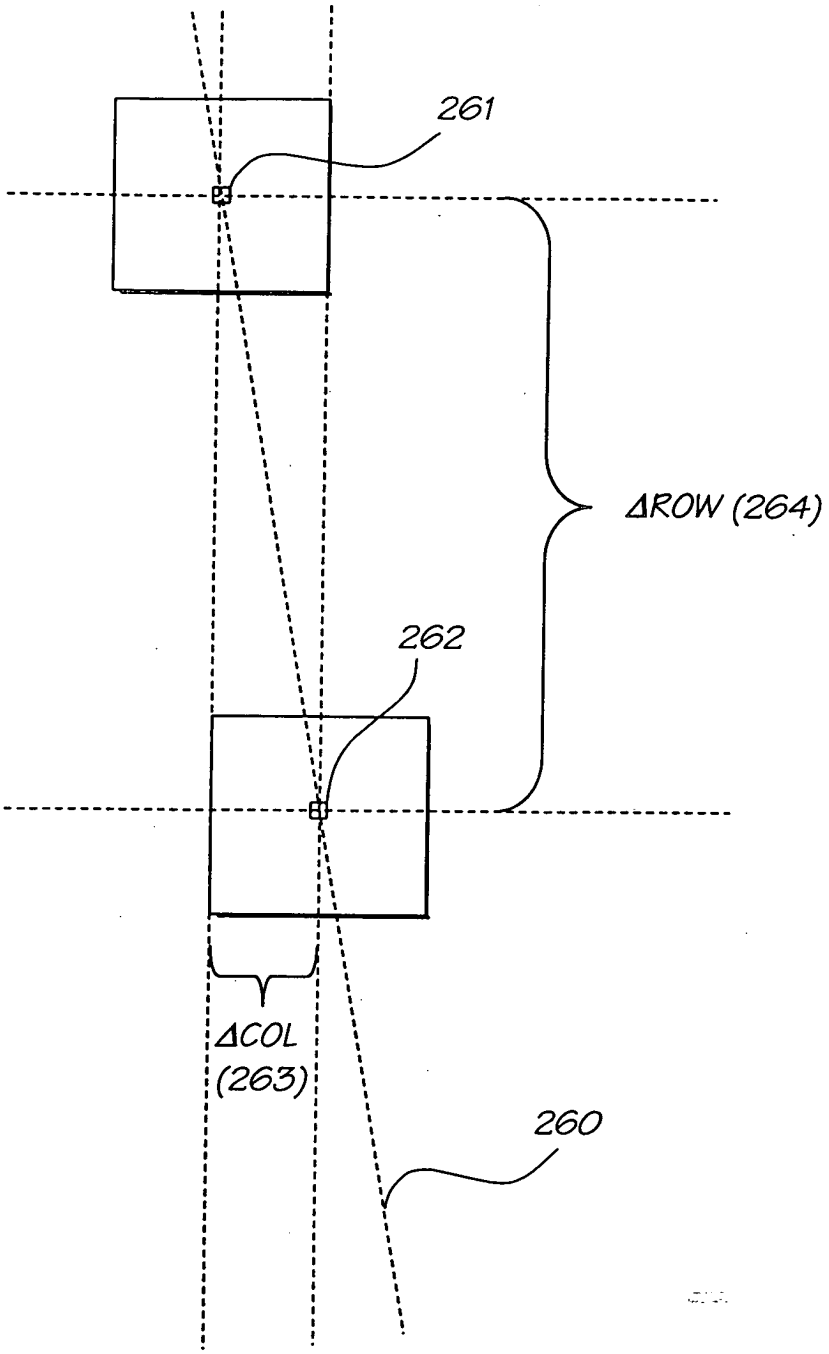


FIG. 41

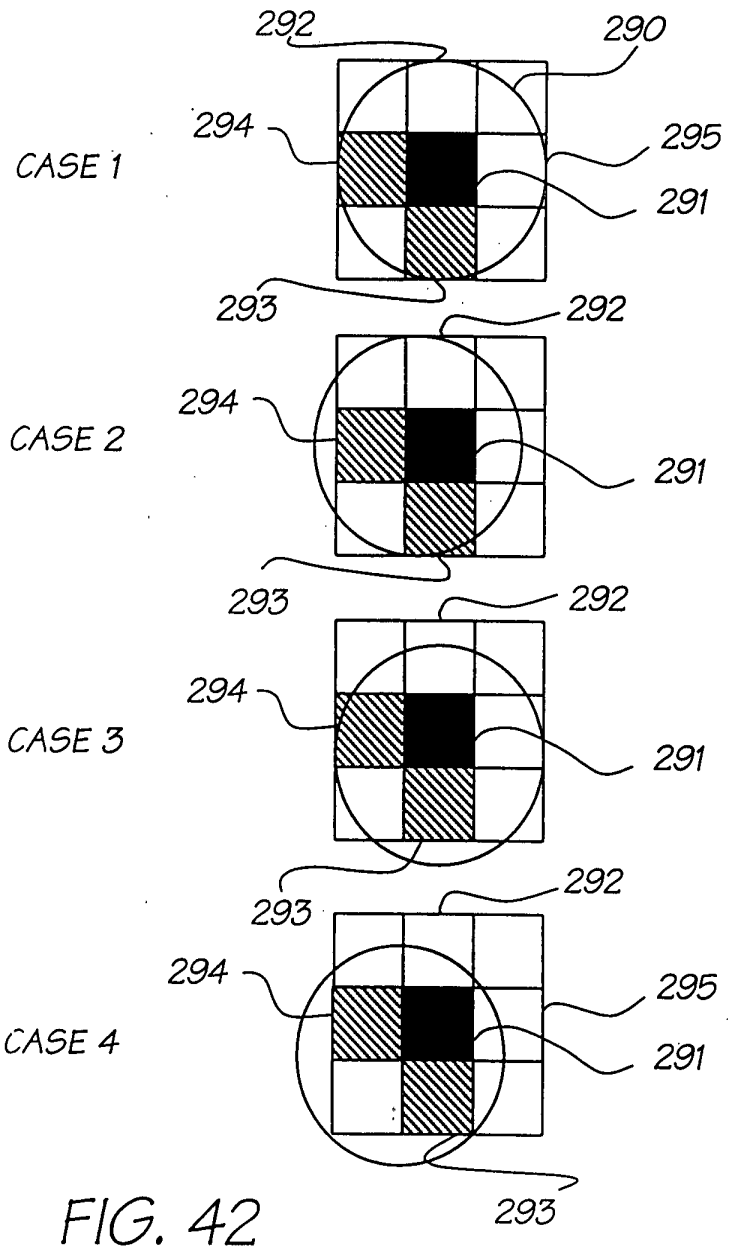

CASE 1
CASE 2
CASE 3
CASE 4
292
290
294
295
291
293

FIG. 42

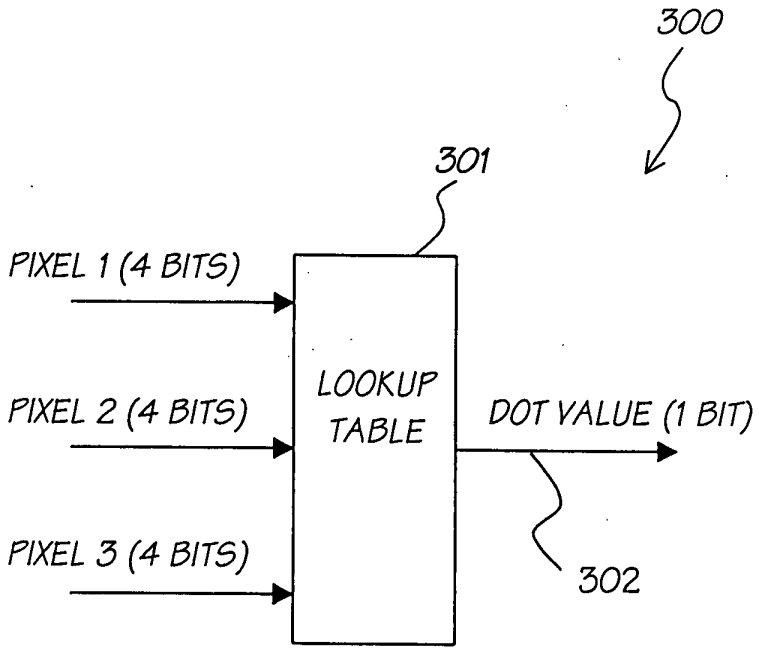

300
301
PIXEL 1 (4 BITS)
PIXEL 2 (4 BITS)
PIXEL 3 (4 BITS)
LOOKUP TABLE
DOT VALUE (1 BIT)
302

FIG. 43

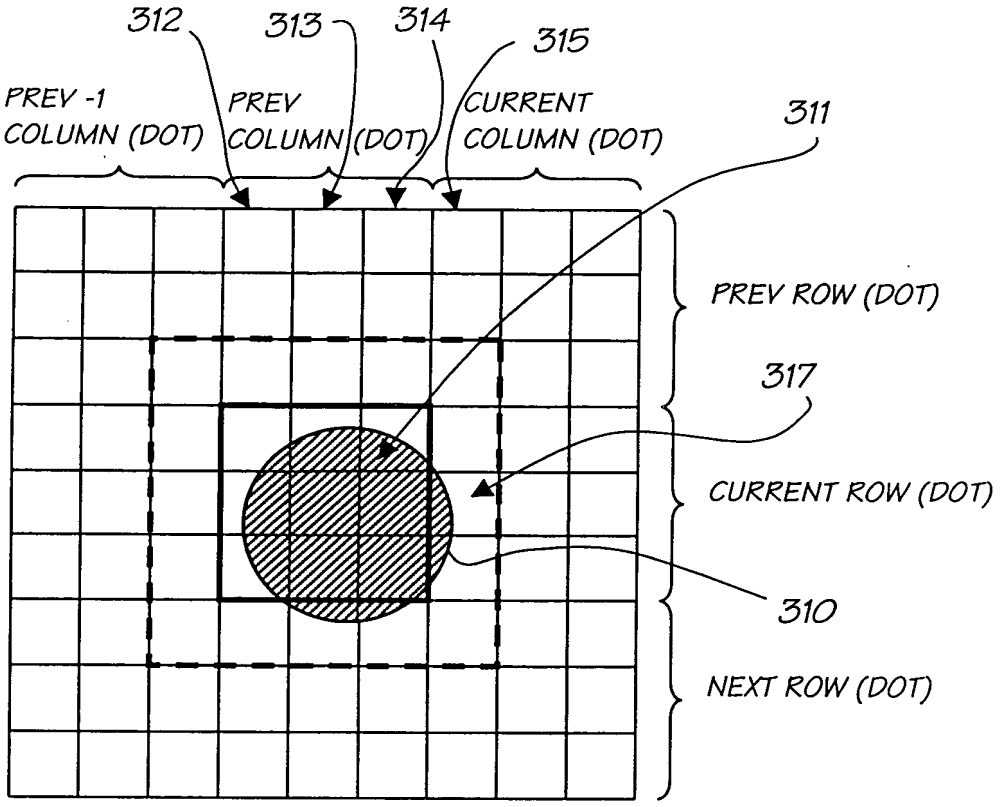

312
313
314
315
PREV -1 COLUMN (DOT)
PREV COLUMN (DOT)
CURRENT COLUMN (DOT)
311
PREV ROW (DOT)
317
CURRENT ROW (DOT)
310
NEXT ROW (DOT)

FIG. 44

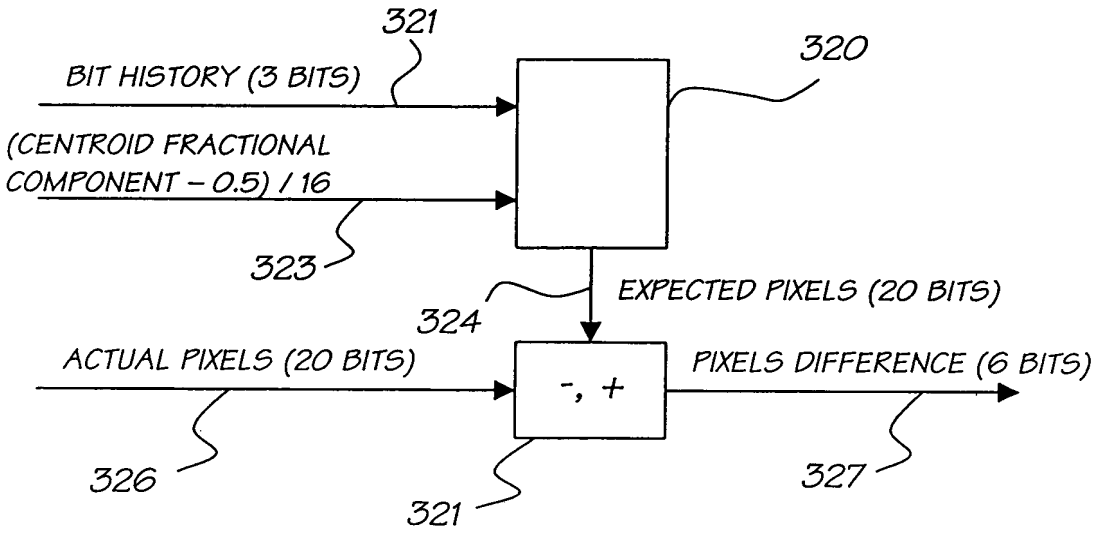

321
BIT HISTORY (3 BITS)
320
(CENTROID FRACTIONAL COMPONENT – 0.5) / 16
323
EXPECTED PIXELS (20 BITS)
324
ACTUAL PIXELS (20 BITS)
-, +
PIXELS DIFFERENCE (6 BITS)
326
321
327

FIG. 45

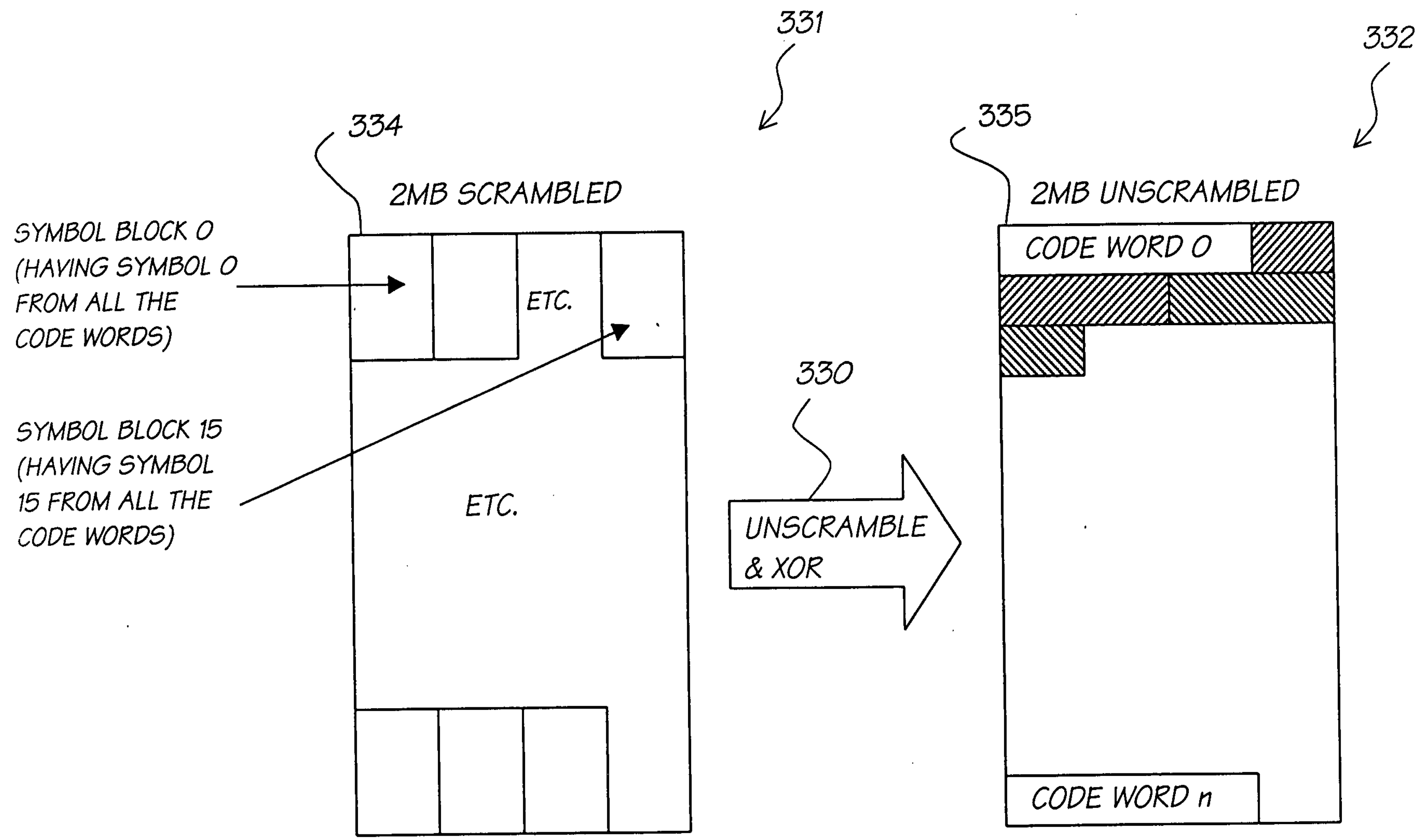

331
332
334
335
2MB SCRAMBLED
2MB UNSCRAMBLED
SYMBOL BLOCK 0 (HAVING SYMBOL 0 FROM ALL THE CODE WORDS)
SYMBOL BLOCK 15 (HAVING SYMBOL 15 FROM ALL THE CODE WORDS)
ETC.
ETC.
330
UNSCRAMBLE & XOR
CODE WORD 0
CODE WORD n

FIG. 46

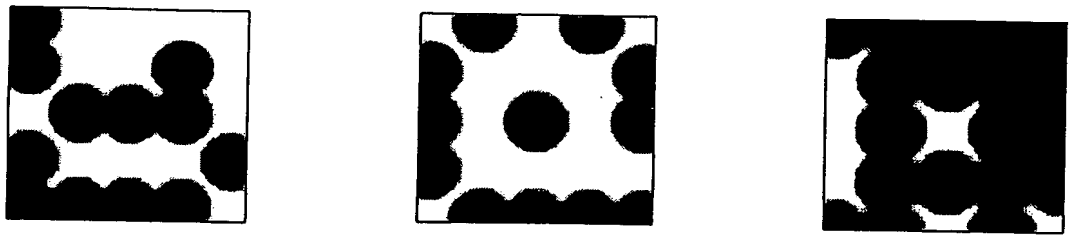

BLACK AND WHITE DOTS
BLACK DOT SURROUNDED BY WHITE
WHITE DOT SURROUNDED BY BLACK

FIG. 47

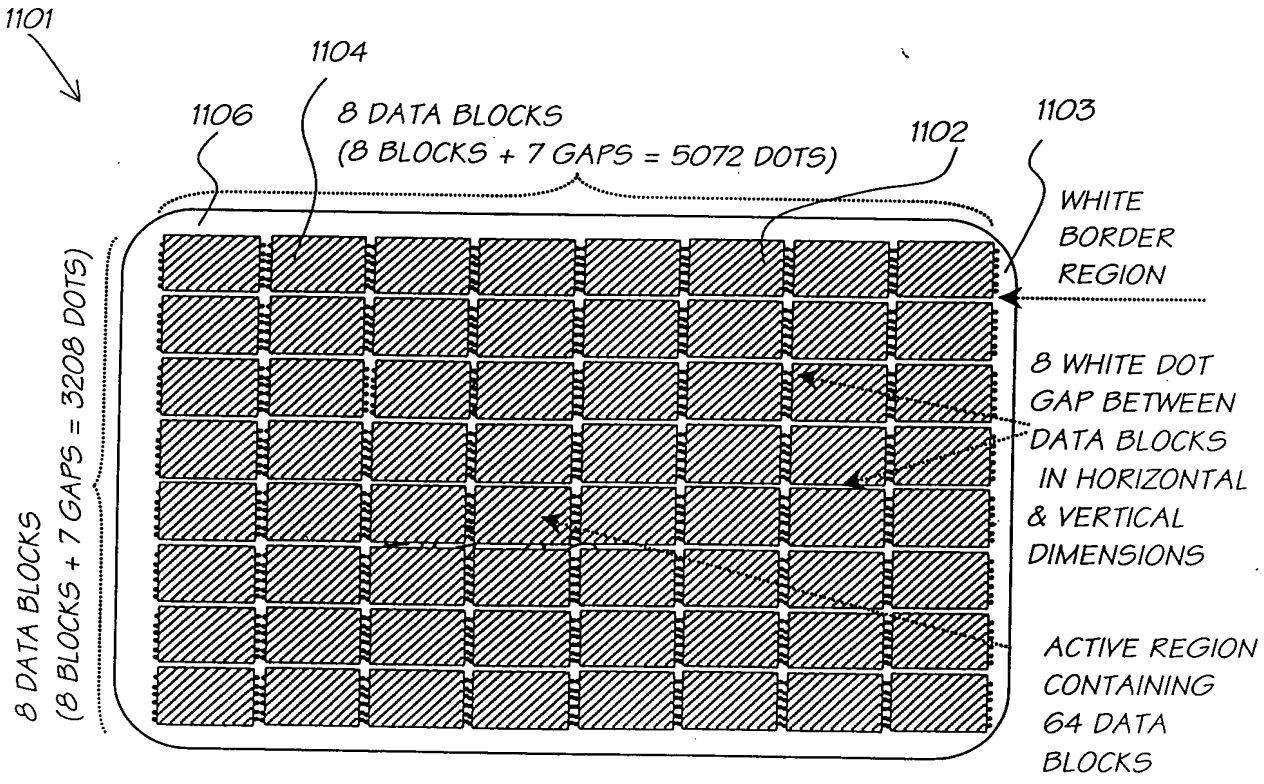

1101
1104
1106
8 DATA BLOCKS
(8 BLOCKS + 7 GAPS = 5072 DOTS)
1102
1103
WHITE BORDER REGION
8 WHITE DOT GAP BETWEEN DATA BLOCKS IN HORIZONTAL & VERTICAL DIMENSIONS
ACTIVE REGION CONTAINING 64 DATA BLOCKS
8 DATA BLOCKS
(8 BLOCKS + 7 GAPS = 3208 DOTS)

FIG. 48

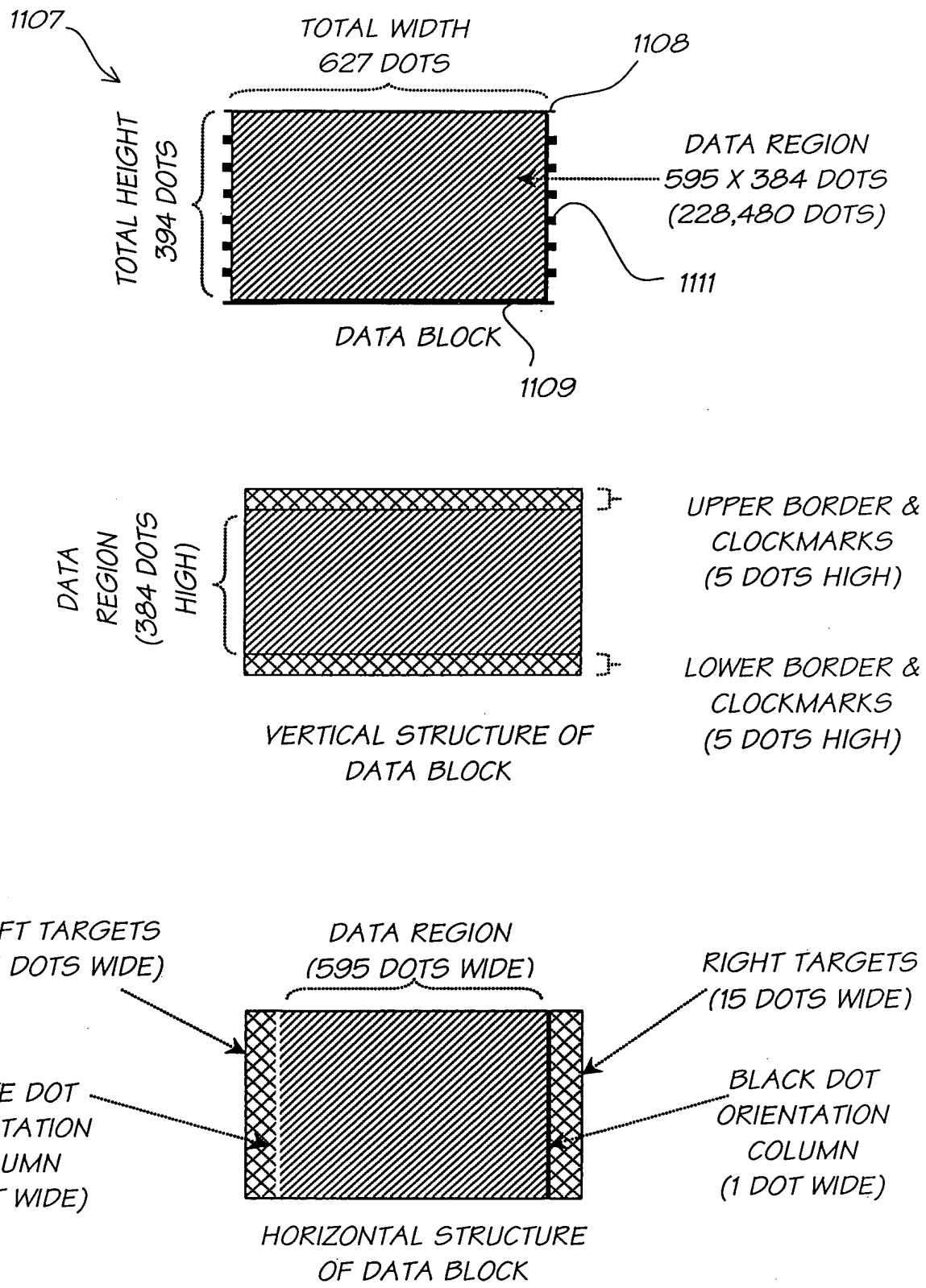

1107
TOTAL WIDTH
627 DOTS
1108
TOTAL HEIGHT
394 DOTS
DATA REGION
595 X 384 DOTS
(228,480 DOTS)
1111
DATA BLOCK
1109
DATA
REGION
(384 DOTS
HIGH)
UPPER BORDER &
CLOCKMARKS
(5 DOTS HIGH)
LOWER BORDER &
CLOCKMARKS
(5 DOTS HIGH)
VERTICAL STRUCTURE OF
DATA BLOCK
LEFT TARGETS
(15 DOTS WIDE)
DATA REGION
(595 DOTS WIDE)
RIGHT TARGETS
(15 DOTS WIDE)
WHITE DOT
ORIENTATION
COLUMN
(1 DOT WIDE)
BLACK DOT
ORIENTATION
COLUMN
(1 DOT WIDE)
HORIZONTAL STRUCTURE
OF DATA BLOCK

FIG. 49

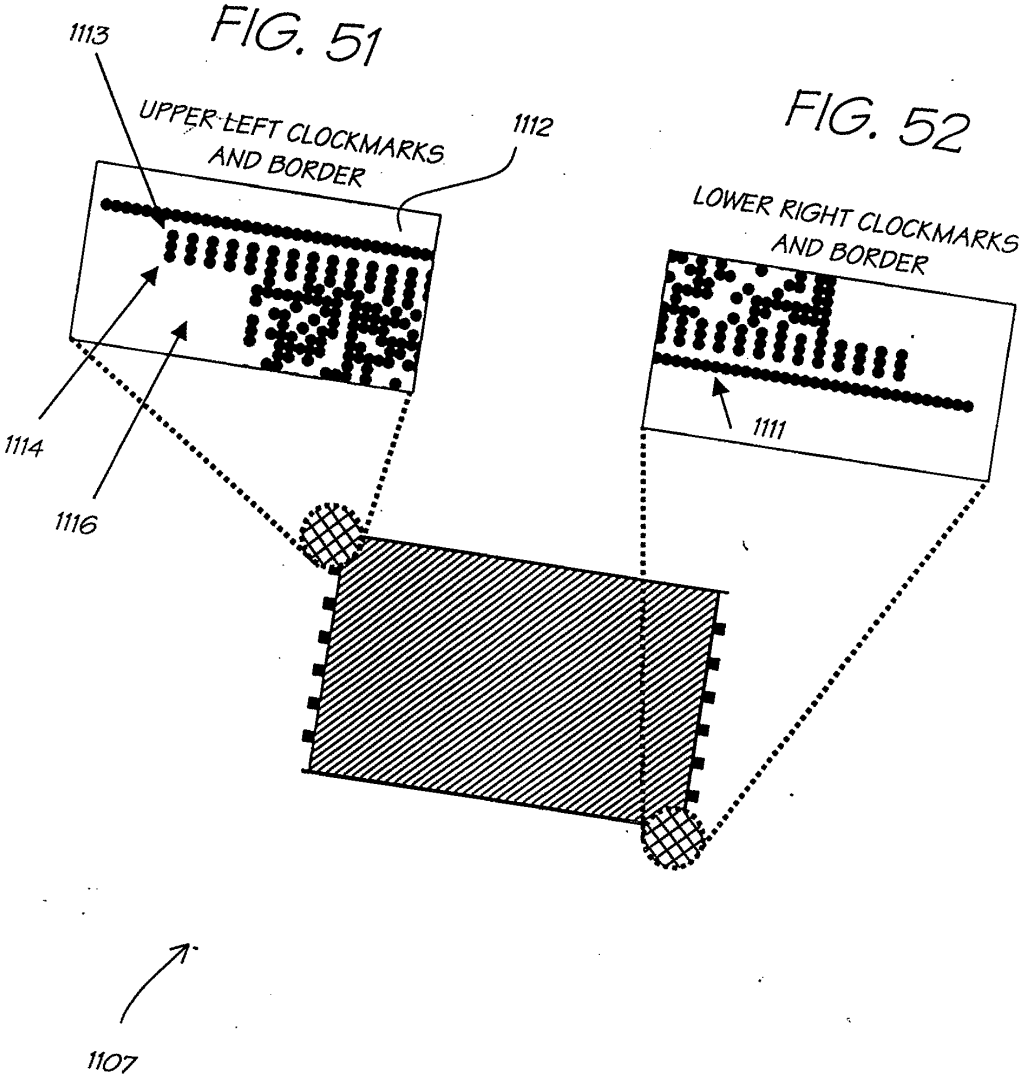


FIG. 51

FIG. 52

FIG. 50

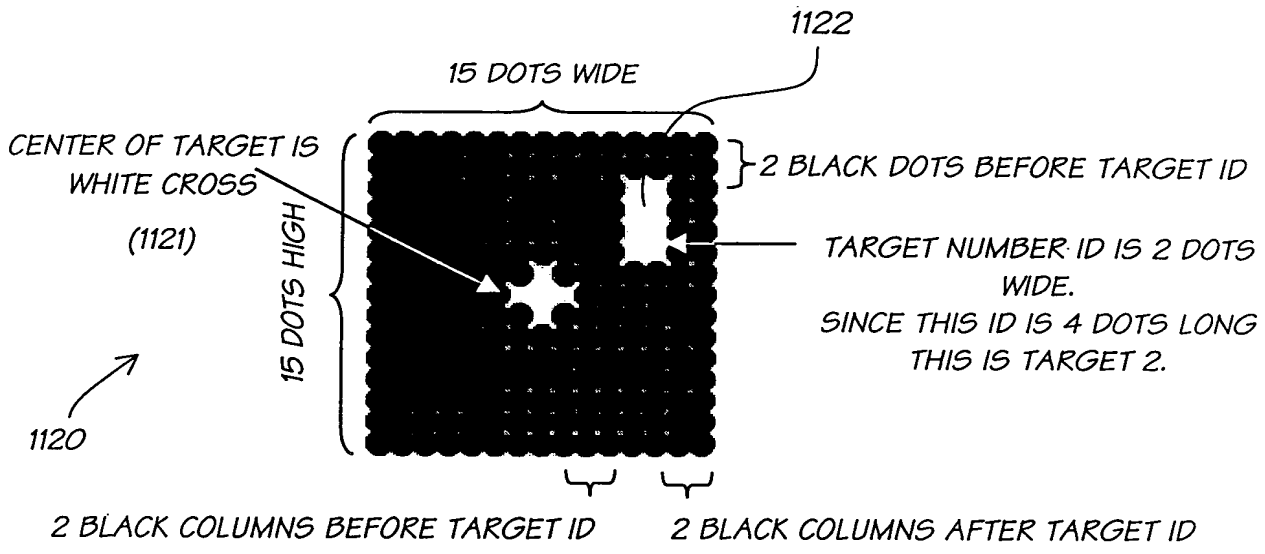

1122
15 DOTS WIDE
CENTER OF TARGET IS WHITE CROSS
(1121)
15 DOTS HIGH
2 BLACK DOTS BEFORE TARGET ID
TARGET NUMBER ID IS 2 DOTS WIDE.
SINCE THIS ID IS 4 DOTS LONG THIS IS TARGET 2.
1120
2 BLACK COLUMNS BEFORE TARGET ID
2 BLACK COLUMNS AFTER TARGET ID

FIG. 53

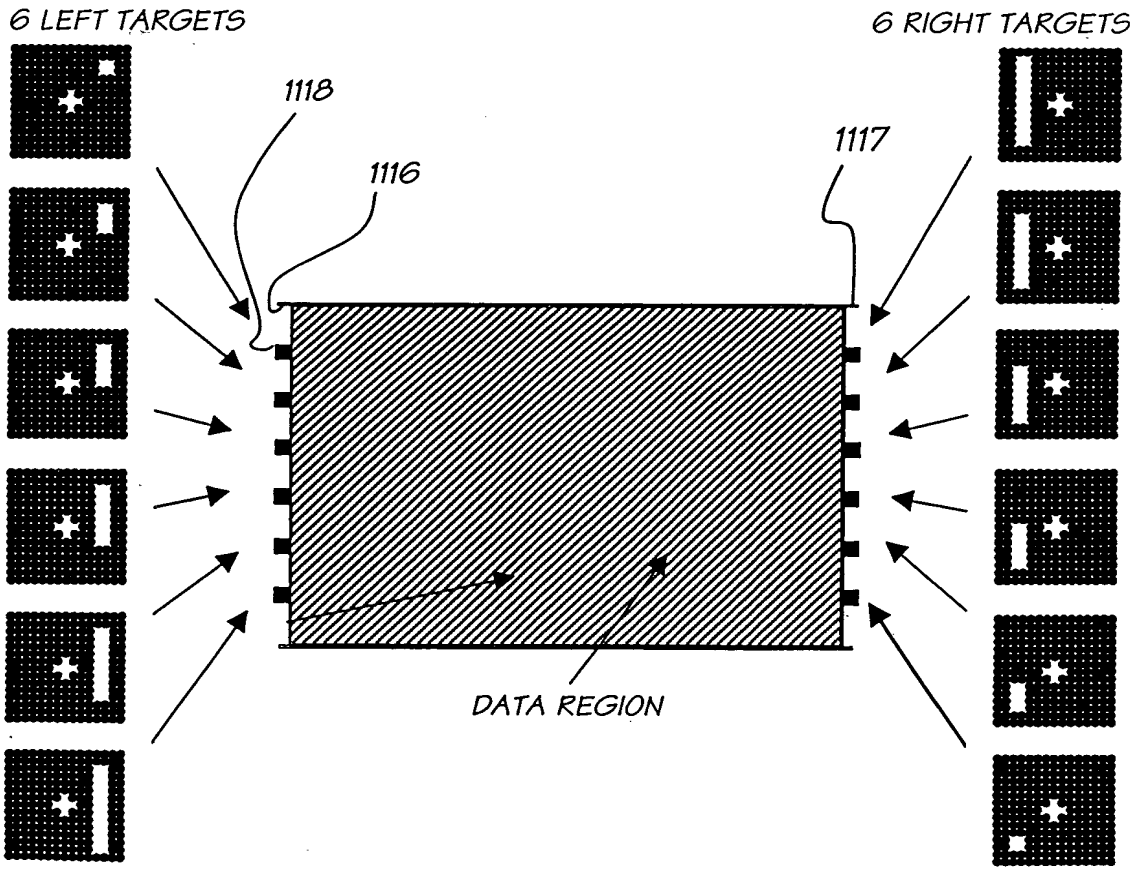

6 LEFT TARGETS
6 RIGHT TARGETS
1118
1116
1117
DATA REGION

FIG. 54

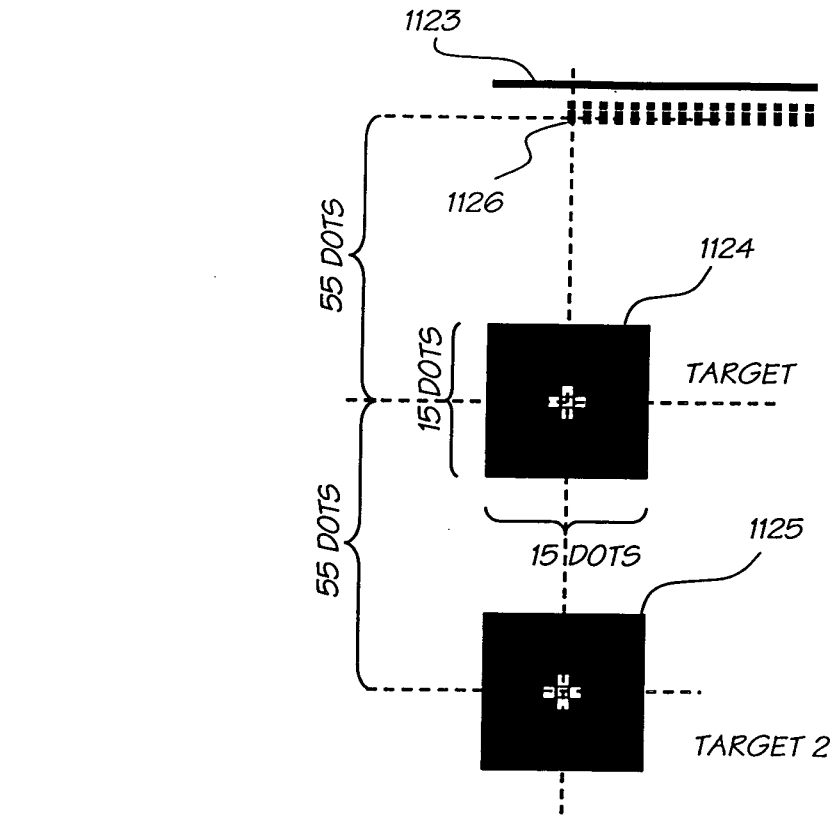

1123
1126
55 DOTS
1124
15 DOTS
TARGET
55 DOTS
15 DOTS
1125
TARGET 2

FIG. 55

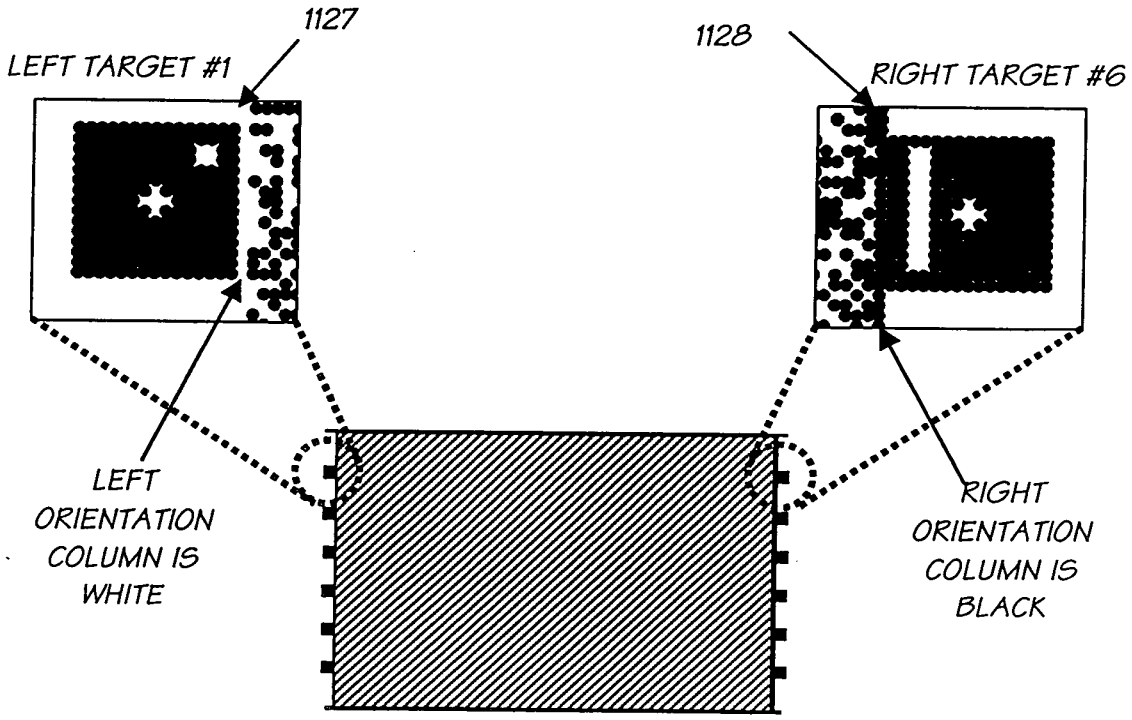

1127
LEFT TARGET #1
1128
RIGHT TARGET #6
LEFT ORIENTATION COLUMN IS WHITE
RIGHT ORIENTATION COLUMN IS BLACK

FIG. 56

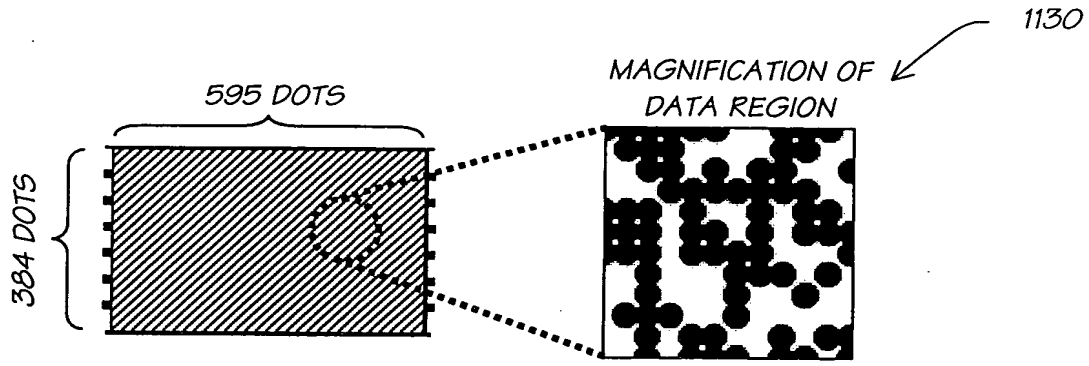


FIG. 57

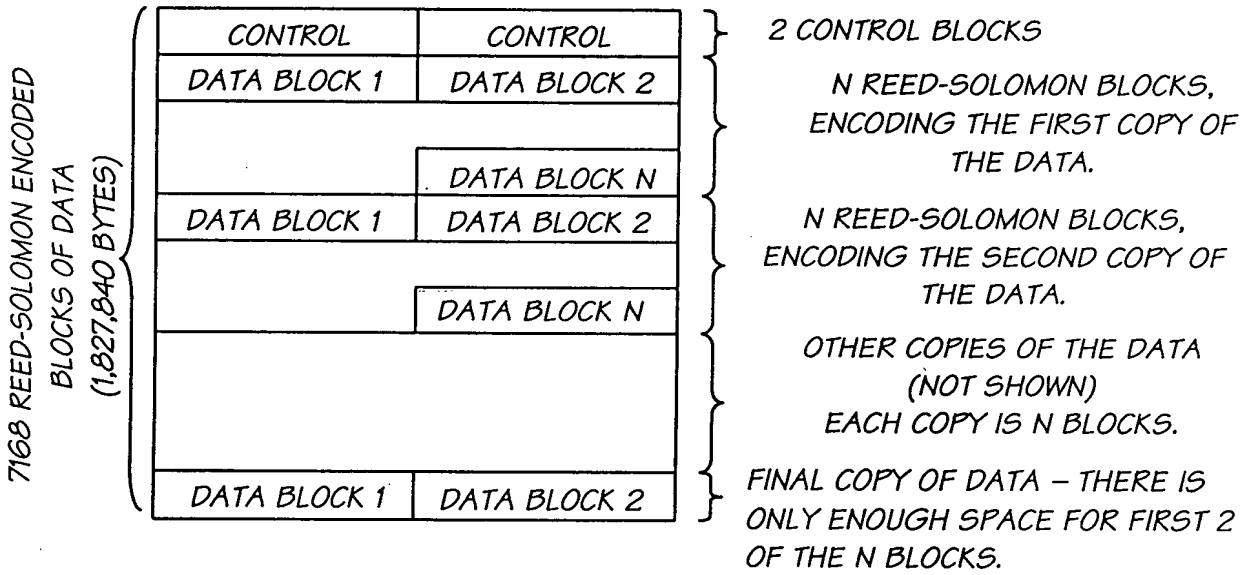


FIG. 58

```
00: 4F 00 3D 4F 00 3D 4F 00 3D 4F 00 3D
0C: 4F 00 3D 4F 00 3D 4F 00 3D 4F 00 3D
18: 4F 00 3D 4F 00 3D 4F 00 3D 4F 00 3D
24: 4F 00 3D 4F 00 3D 4F 00 3D 4F 00 3D
30: 4F 00 3D 4F 00 3D 4F 00 3D 4F 00 3D
3C: 4F 00 3D 4F 00 3D 4F 00 3D 4F 00 3D
48: 4F 00 3D 4F 00 3D 4F 00 3D 4F 00 3D
54: 4F 00 3D 4F 00 3D 4F 00 3D 4F 00 3D
60: 00 00 00 00 00 00 00 00 00 00 00 00
6C: 00 00 00 00 00 00 00 00 00 00 00 00
78: 00 00 00 00 00 00 00 00 00 00 00 00
```

32 COPIES OF THE 3 BYTE CONTROL INFORMATION

RESERVED BYTES ARE 0

FIG. 59

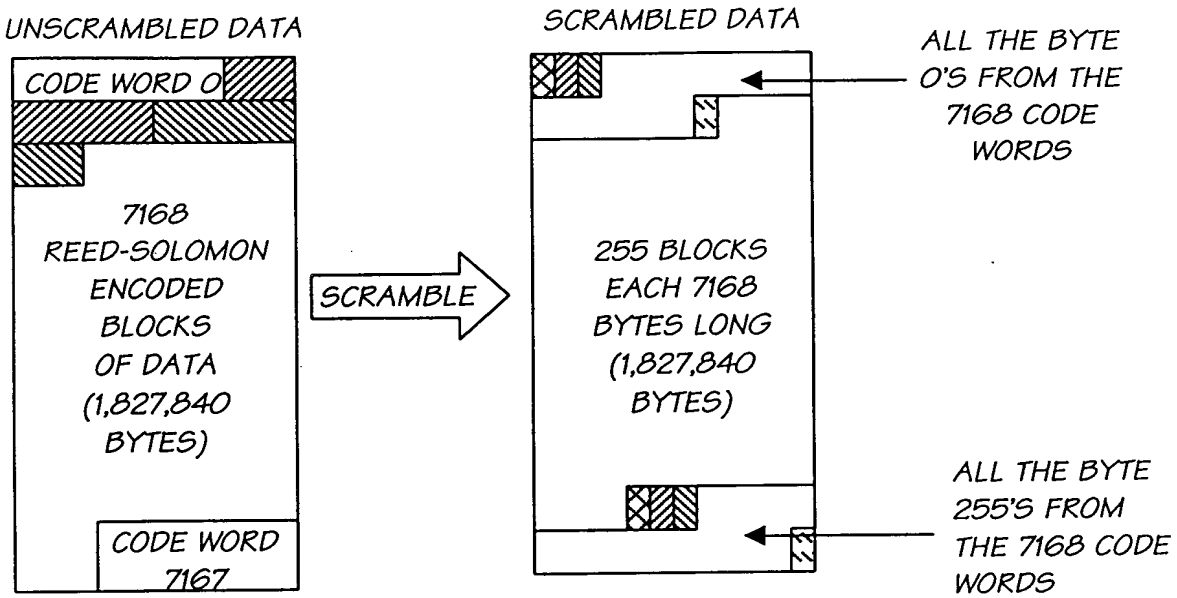


FIG. 60

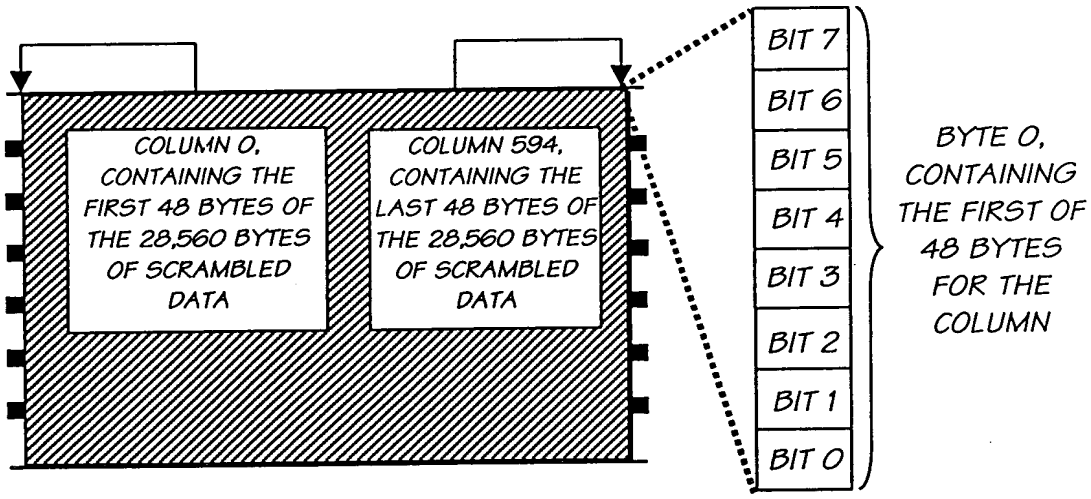


FIG. 61

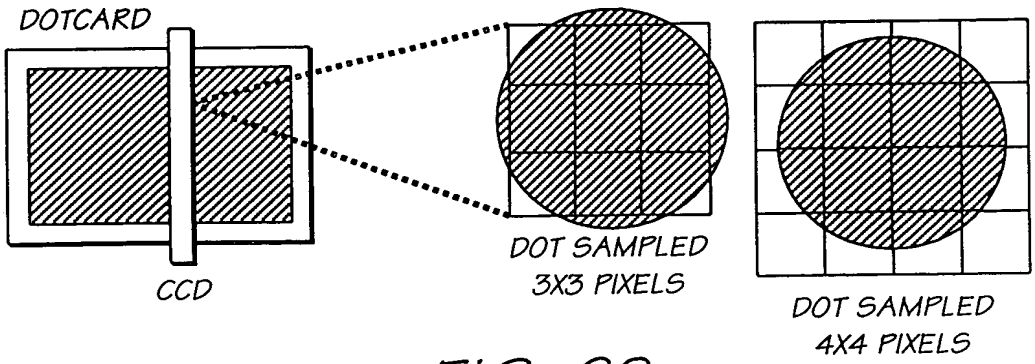


FIG. 62

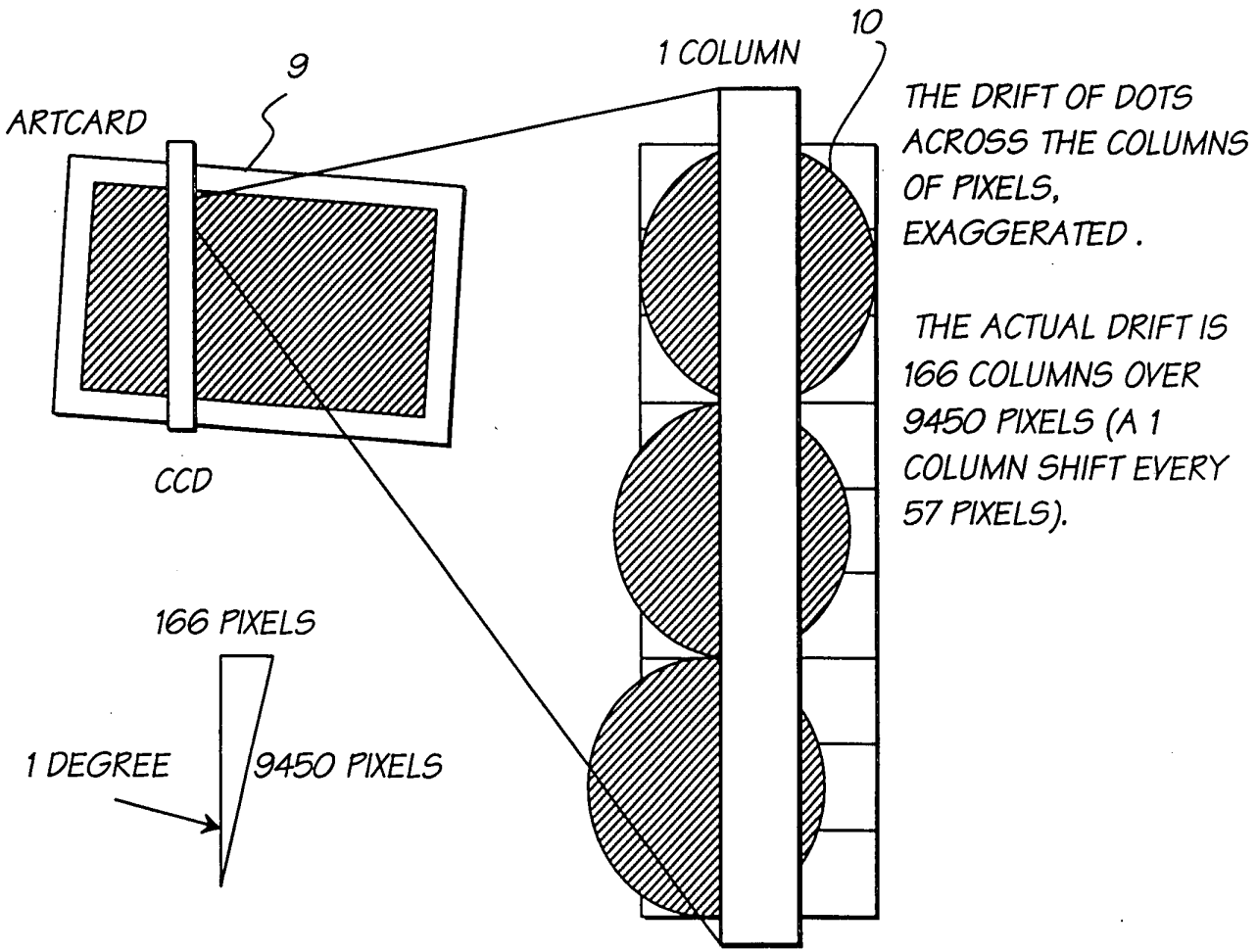

9
ARTCARD
CCD
1 COLUMN
10
THE DRIFT OF DOTS ACROSS THE COLUMNS OF PIXELS, EXAGGERATED .
THE ACTUAL DRIFT IS 166 COLUMNS OVER 9450 PIXELS (A 1 COLUMN SHIFT EVERY 57 PIXELS).
166 PIXELS
1 DEGREE
9450 PIXELS

FIG. 63

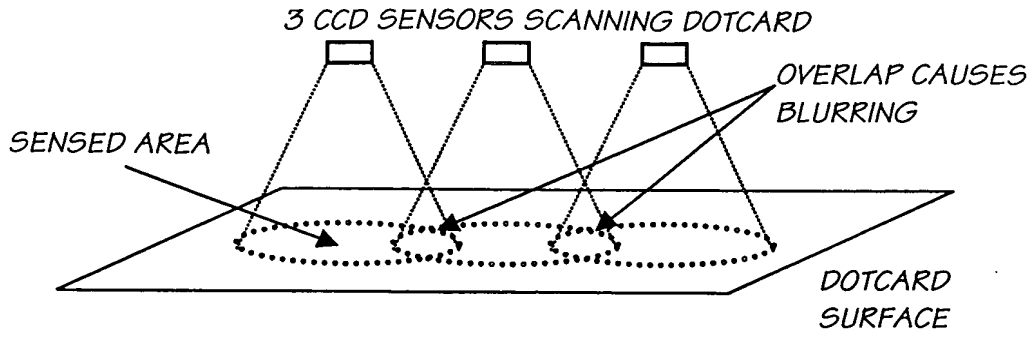


FIG. 64

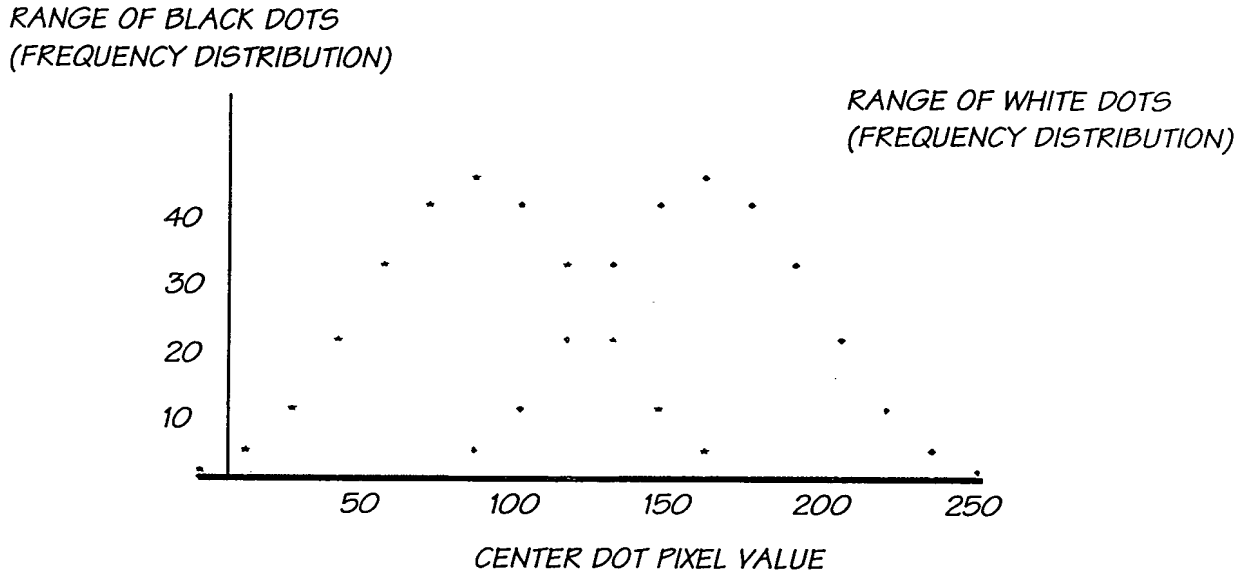


FIG. 65

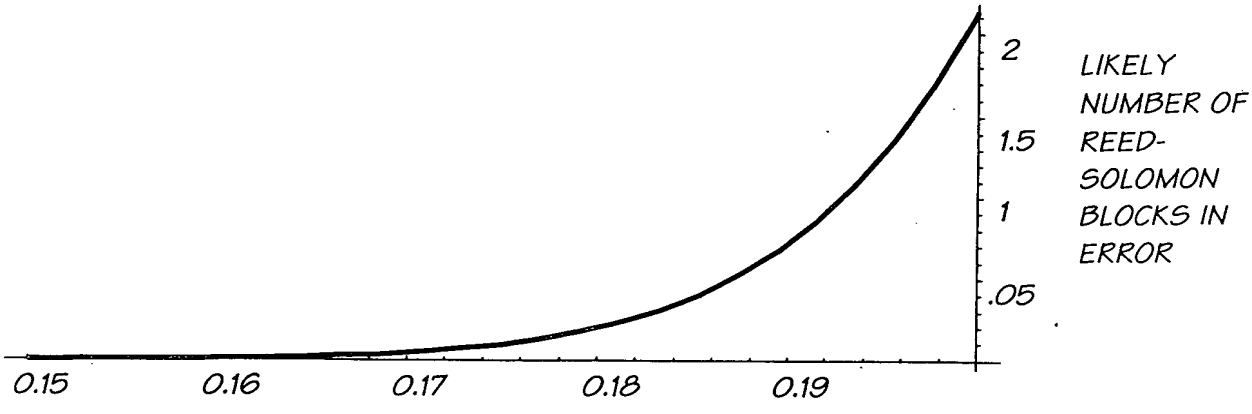
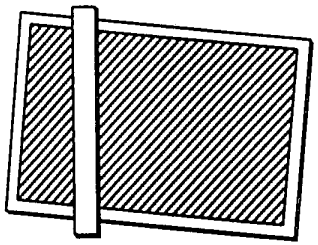


FIG. 66



READ
LINEAR CCD

DOTCARD PIXEL DATA

EXTRACT
BIT IMAGE

ENCODED, SCRAMBLED,
POTENTIALLY ROTATED 180°
BITMAPPED DATA

ROTATE AND
UNSCRAMBLE

ENCODED,
UNSCRAMBLED DATA

REED-
SOLOMON
DECODE

RAW DATA

APPROXIMATE DATA SIZES FOR 1600 DPI DOTCARD

86MM + 1MM IN HORIZONTAL DIMENSION FOR 1° ROTATION = 87MM
87MM = 16,252 SCANLINES
16,440 SCANLINES @ 11,000 PIXELS PER SCANLINE = 180,840,000 PIXELS
180,840,000 PIXELS @ 1 BYTE PER PIXEL = 180,840,000 BYTES = 172.5 MB

64 DATA BLOCKS, EACH CONTAINING 597 COLUMNS (595 DATA REGION COLUMNS AND 2 ORIENTATION COLUMNS), @ 48 BYTES PER COLUMN = 28,656 BYTES PER DATA BLOCK FOR A TOTAL OF 1,833,984 BYTES.

64 DATA BLOCKS, EACH CONTAINING 112 ENCODED REED SOLOMON BLOCKS, @ 255 BYTES PER REED SOLOMON BLOCK FOR A TOTAL OF 1,827,840 BYTES.

DECODED DATA, WITH A MAXIMUM SIZE OF 910,082 BYTES.
(64 X 112 X 127 – (2 CONTROL BLOCKS @ 127 BYTES))

FIG. 67

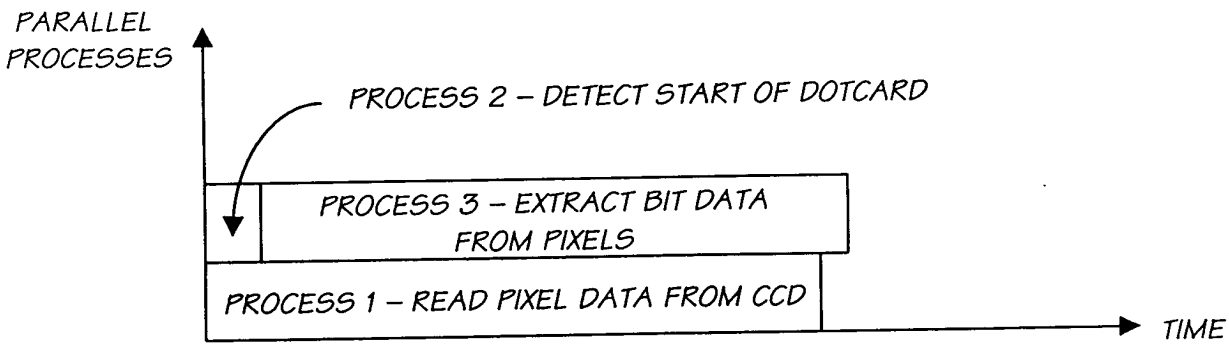

PARALLEL PROCESSES
PROCESS 2 – DETECT START OF DOTCARD
PROCESS 3 – EXTRACT BIT DATA FROM PIXELS
PROCESS 1 – READ PIXEL DATA FROM CCD
TIME

FIG. 68

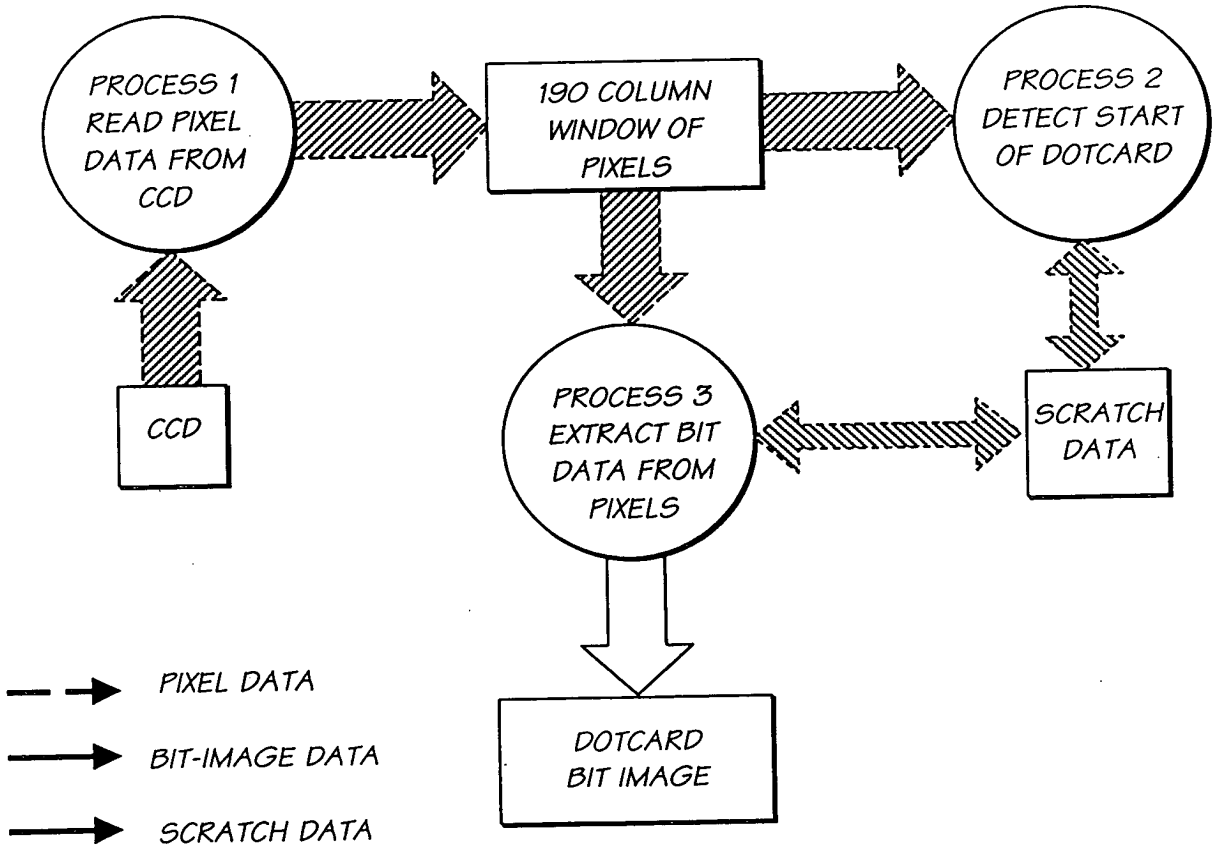

PROCESS 1 READ PIXEL DATA FROM CCD
190 COLUMN WINDOW OF PIXELS
PROCESS 2 DETECT START OF DOTCARD
CCD
PROCESS 3 EXTRACT BIT DATA FROM PIXELS
SCRATCH DATA
DOTCARD BIT IMAGE
PIXEL DATA
BIT-IMAGE DATA
SCRATCH DATA

FIG. 69

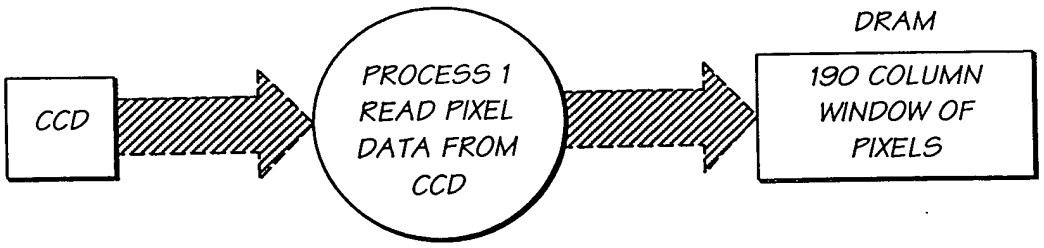


FIG. 70

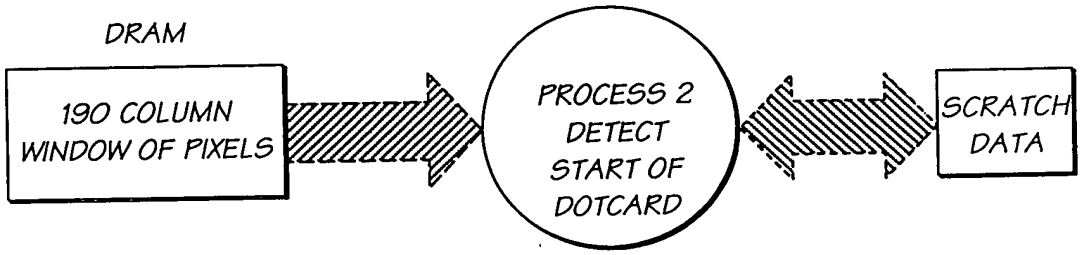


FIG. 71

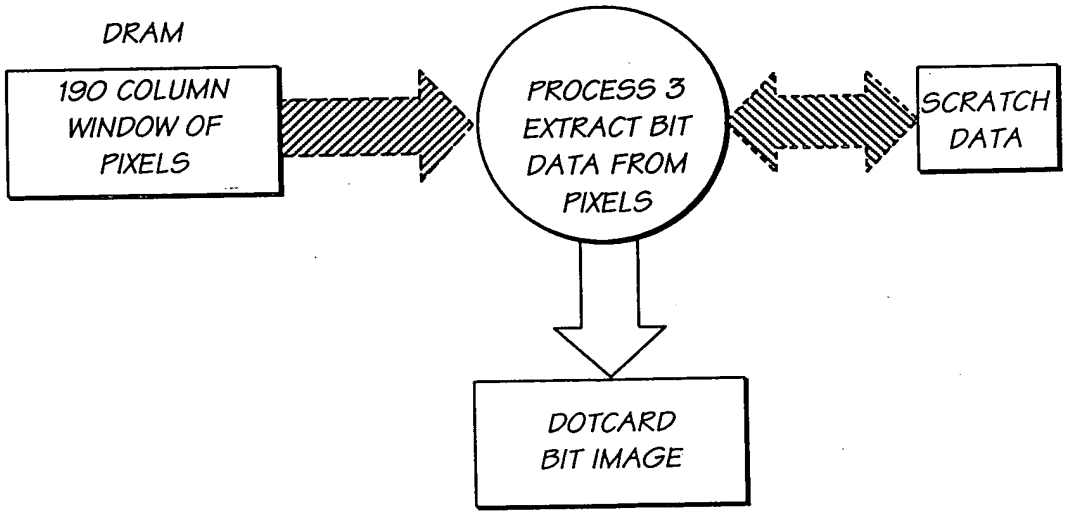


FIG. 72

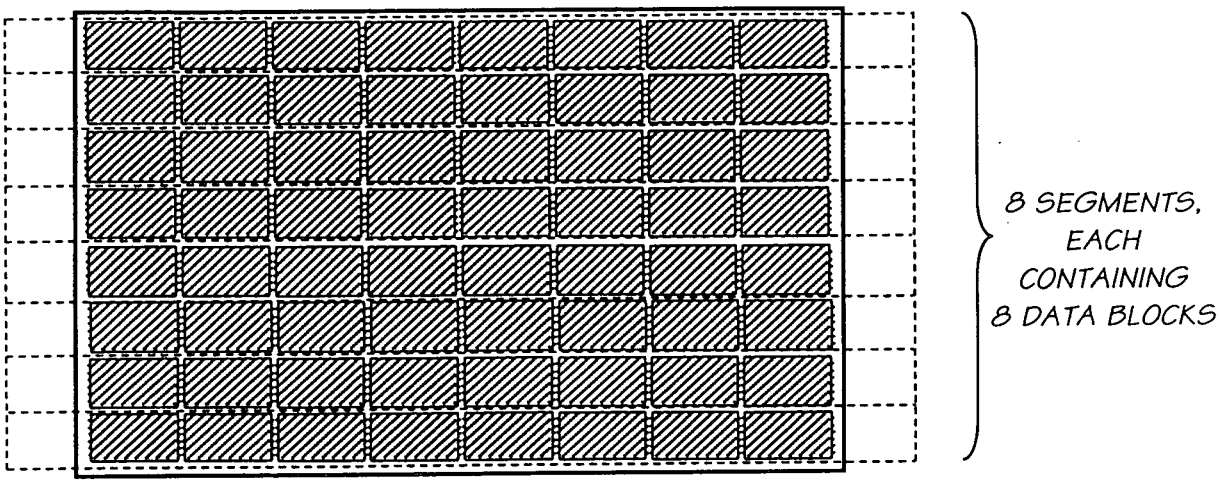

8 SEGMENTS,
EACH
CONTAINING
8 DATA BLOCKS

FIG. 73

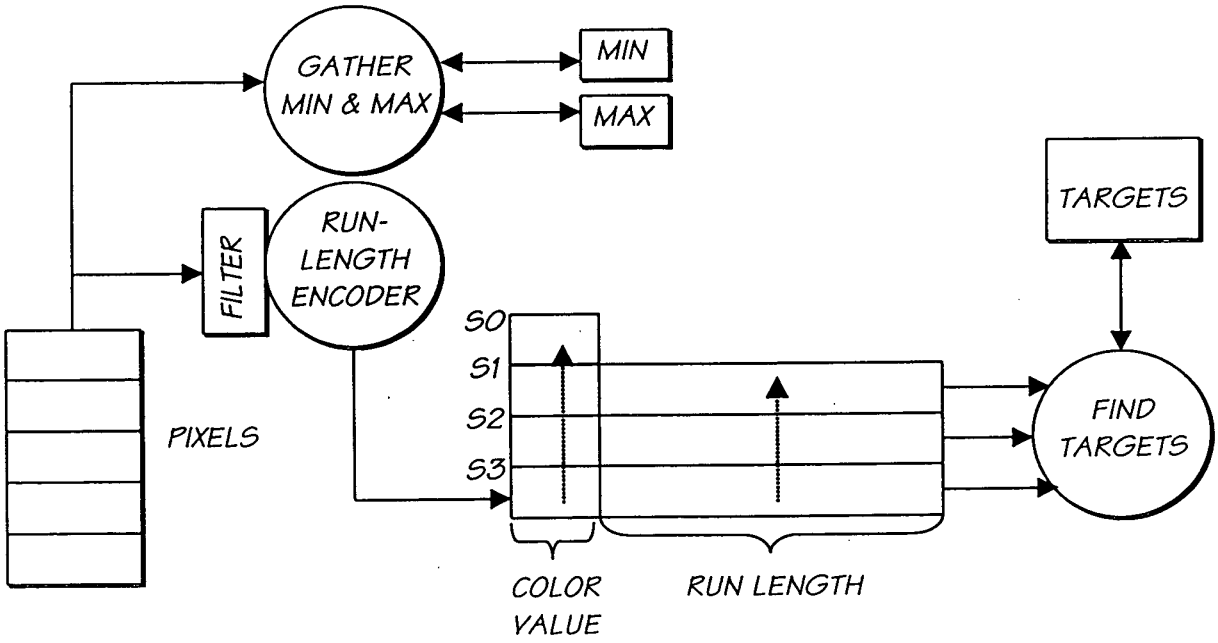

GATHER
MIN & MAX
MIN
MAX
FILTER
RUN-
LENGTH
ENCODER
TARGETS
PIXELS
S0
S1
S2
S3
FIND
TARGETS
COLOR
VALUE
RUN LENGTH

FIG. 74

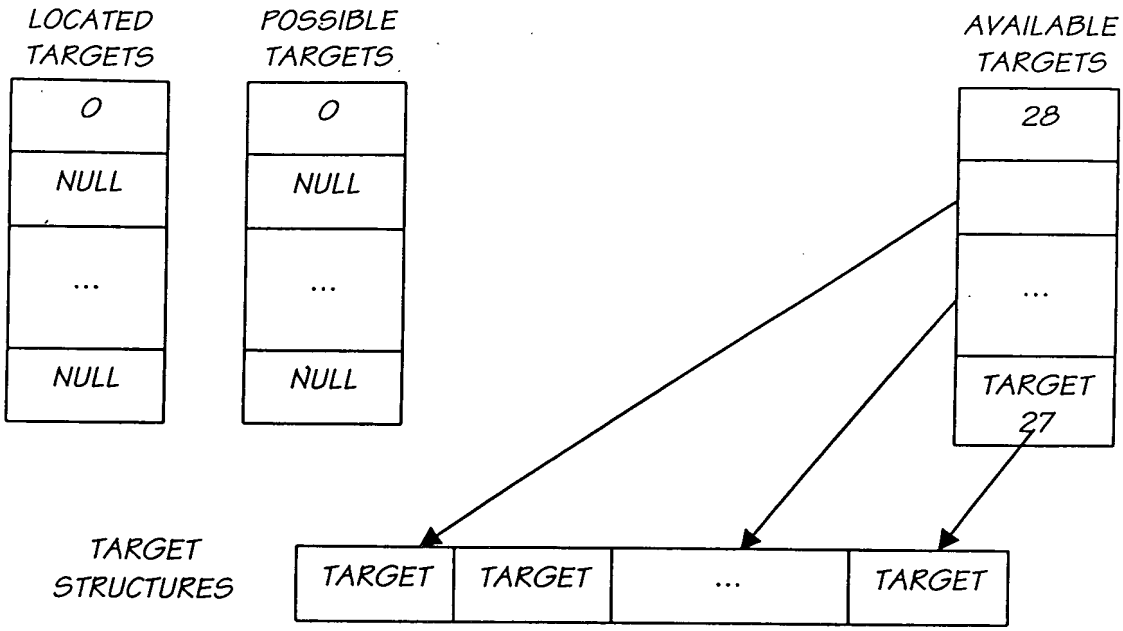
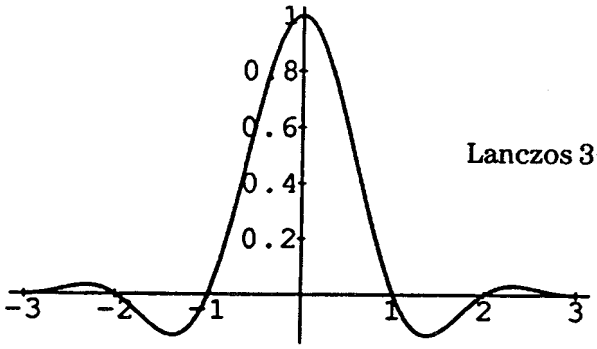


FIG. 75



$$\text{Lanczos}\,3(x) = \begin{cases} \dfrac{\sin(\pi x)}{\pi x}\,\dfrac{\sin\left(\pi\dfrac{x}{3}\right)}{\pi\dfrac{x}{3}}, & |x| < 3 \\ 0, & |x| \geq 3 \end{cases}$$

FIG. 76

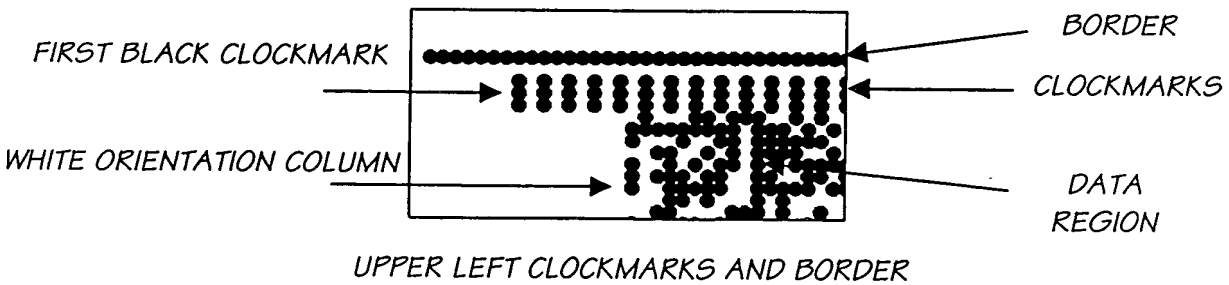


FIG. 77

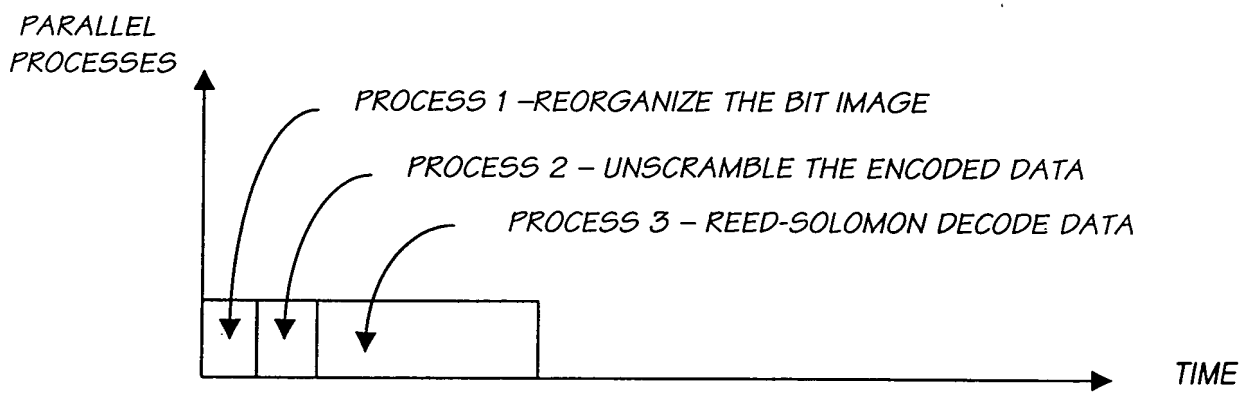

PARALLEL
PROCESSES
PROCESS 1 –REORGANIZE THE BIT IMAGE
PROCESS 2 – UNSCRAMBLE THE ENCODED DATA
PROCESS 3 – REED-SOLOMON DECODE DATA
TIME

FIG. 78

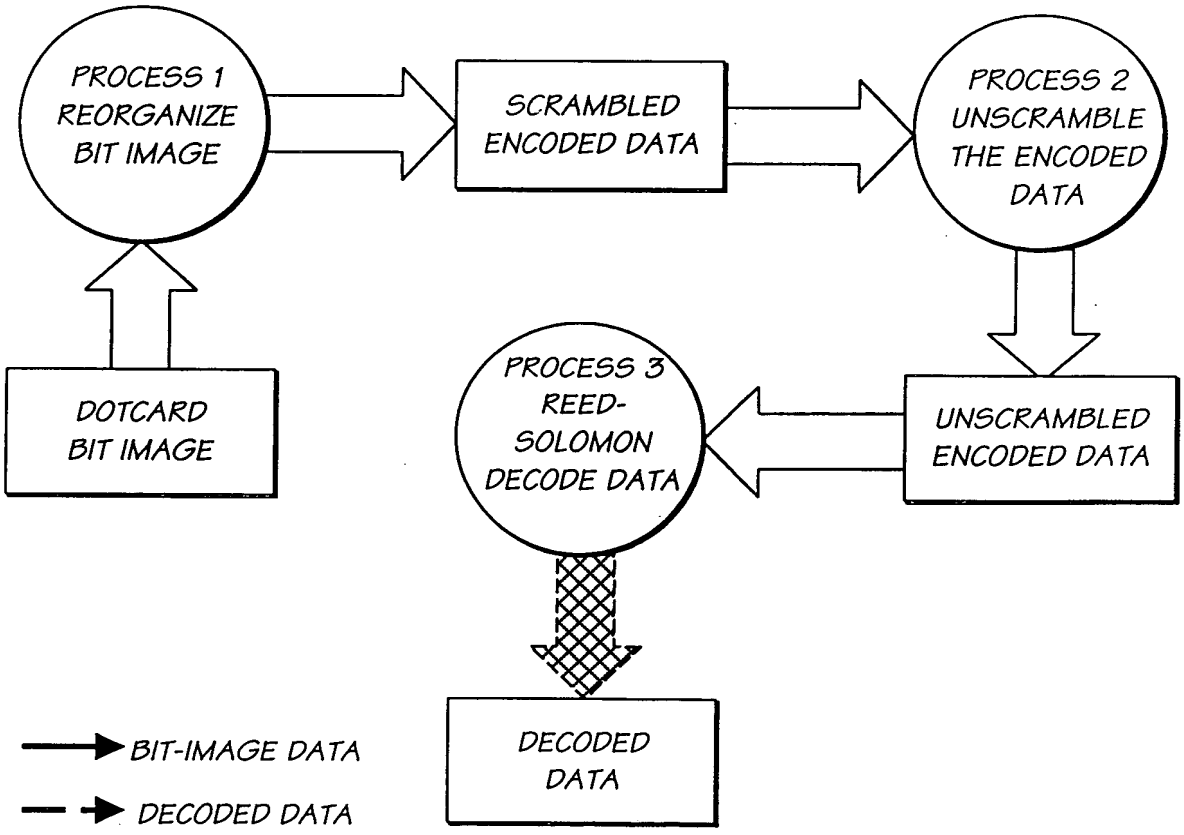

PROCESS 1
REORGANIZE
BIT IMAGE
SCRAMBLED
ENCODED DATA
PROCESS 2
UNSCRAMBLE
THE ENCODED
DATA
DOTCARD
BIT IMAGE
PROCESS 3
REED-
SOLOMON
DECODE DATA
UNSCRAMBLED
ENCODED DATA
DECODED
DATA
BIT-IMAGE DATA
DECODED DATA

FIG. 79

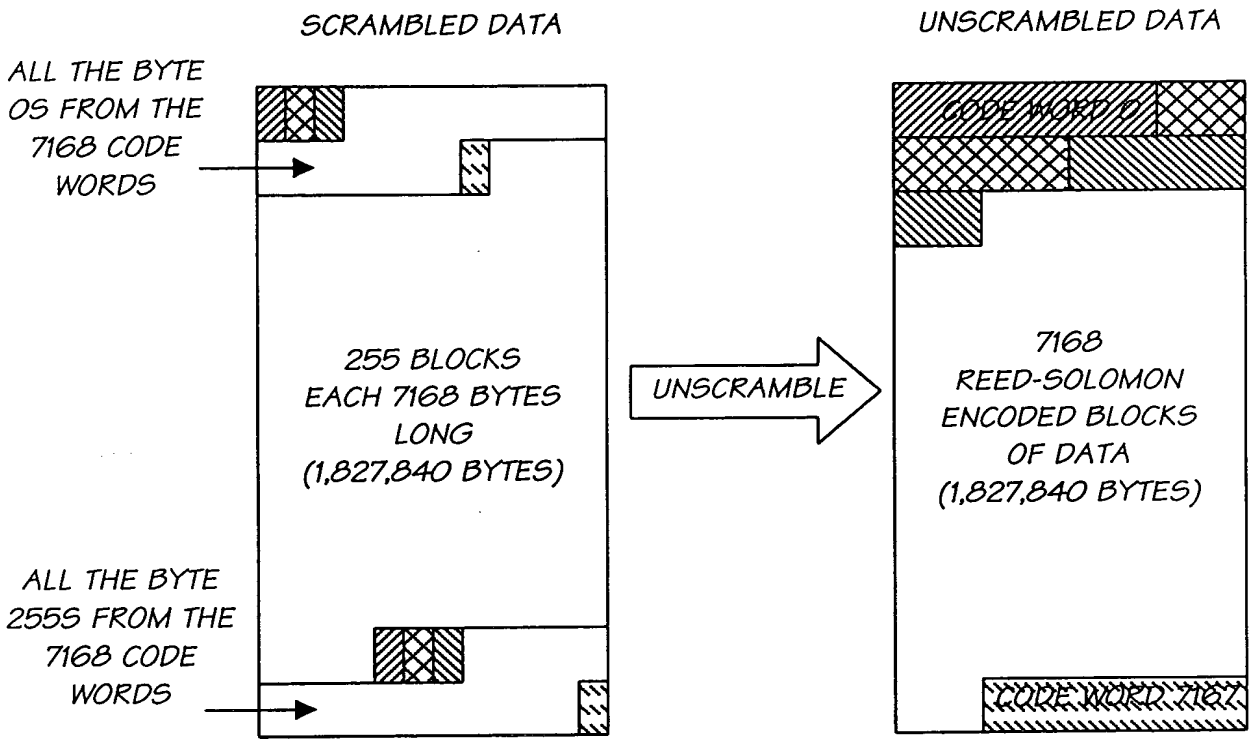

SCRAMBLED DATA
UNSCRAMBLED DATA
ALL THE BYTE OS FROM THE 7168 CODE WORDS
255 BLOCKS EACH 7168 BYTES LONG (1,827,840 BYTES)
ALL THE BYTE 255S FROM THE 7168 CODE WORDS
UNSCRAMBLE
CODE WORD 0
7168 REED-SOLOMON ENCODED BLOCKS OF DATA (1,827,840 BYTES)
CODE WORD 7167

FIG. 80

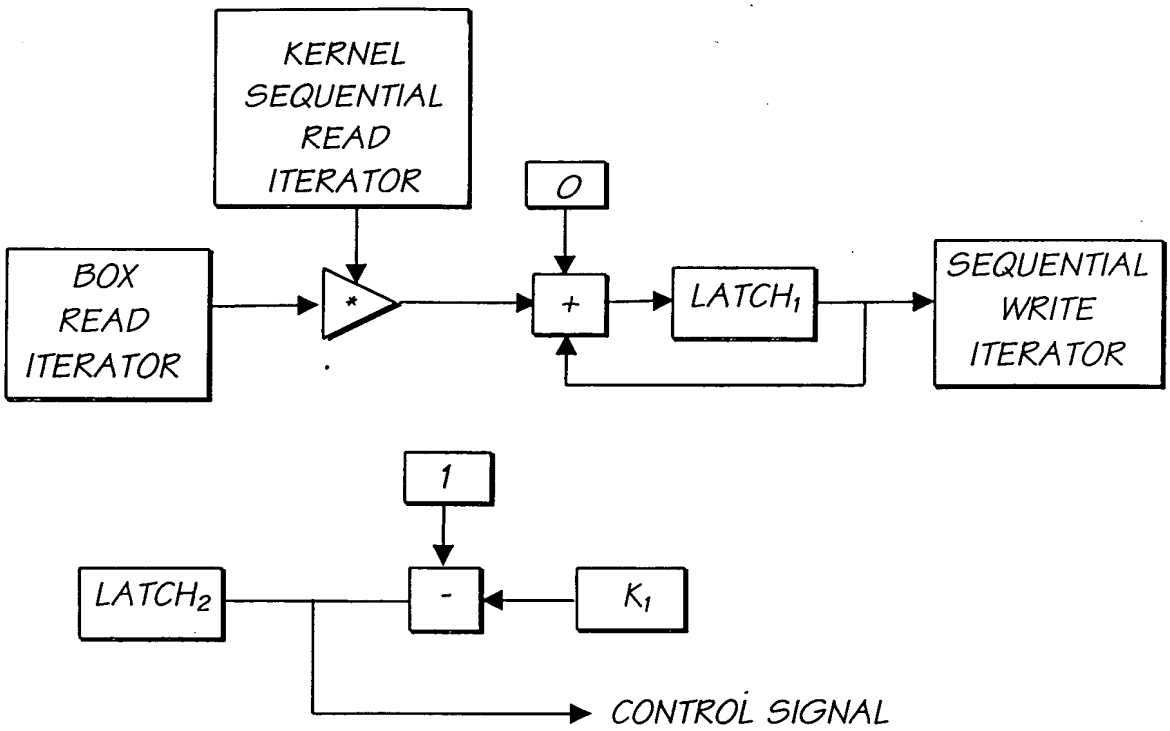

KERNEL SEQUENTIAL READ ITERATOR
0
BOX READ ITERATOR
*
+
LATCH1
SEQUENTIAL WRITE ITERATOR
1
LATCH2
-
K1
CONTROL SIGNAL

FIG. 81

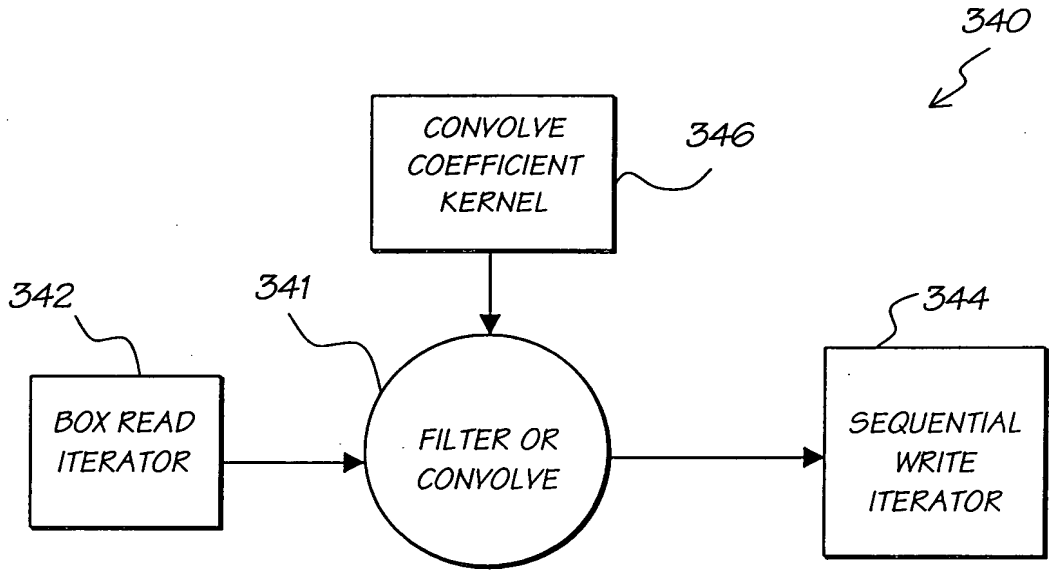

340
CONVOLVE COEFFICIENT KERNEL
346
342
341
344
BOX READ ITERATOR
FILTER OR CONVOLVE
SEQUENTIAL WRITE ITERATOR

FIG. 82

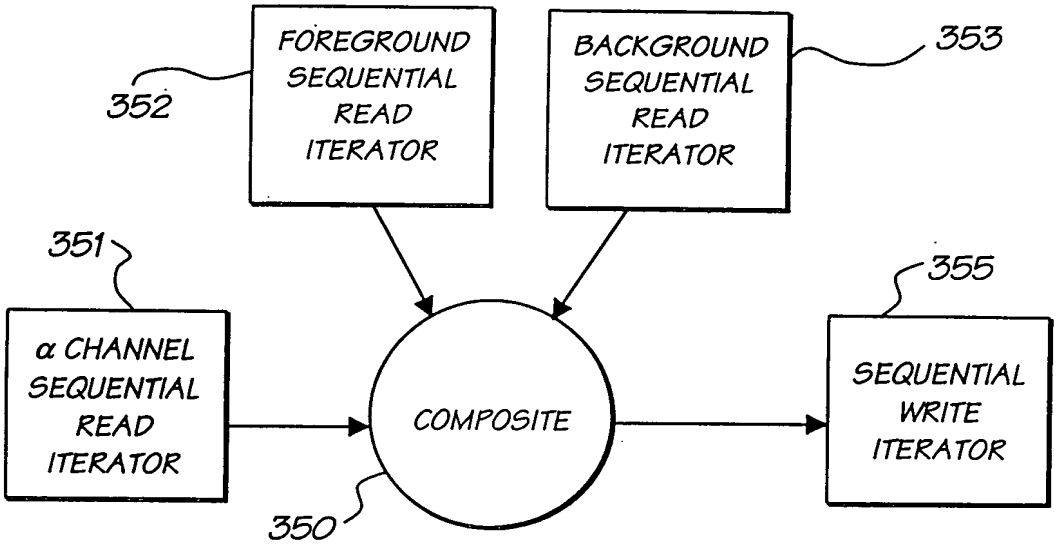

FOREGROUND SEQUENTIAL READ ITERATOR
352
BACKGROUND SEQUENTIAL READ ITERATOR
353
351
α CHANNEL SEQUENTIAL READ ITERATOR
COMPOSITE
355
SEQUENTIAL WRITE ITERATOR
350

FIG. 83

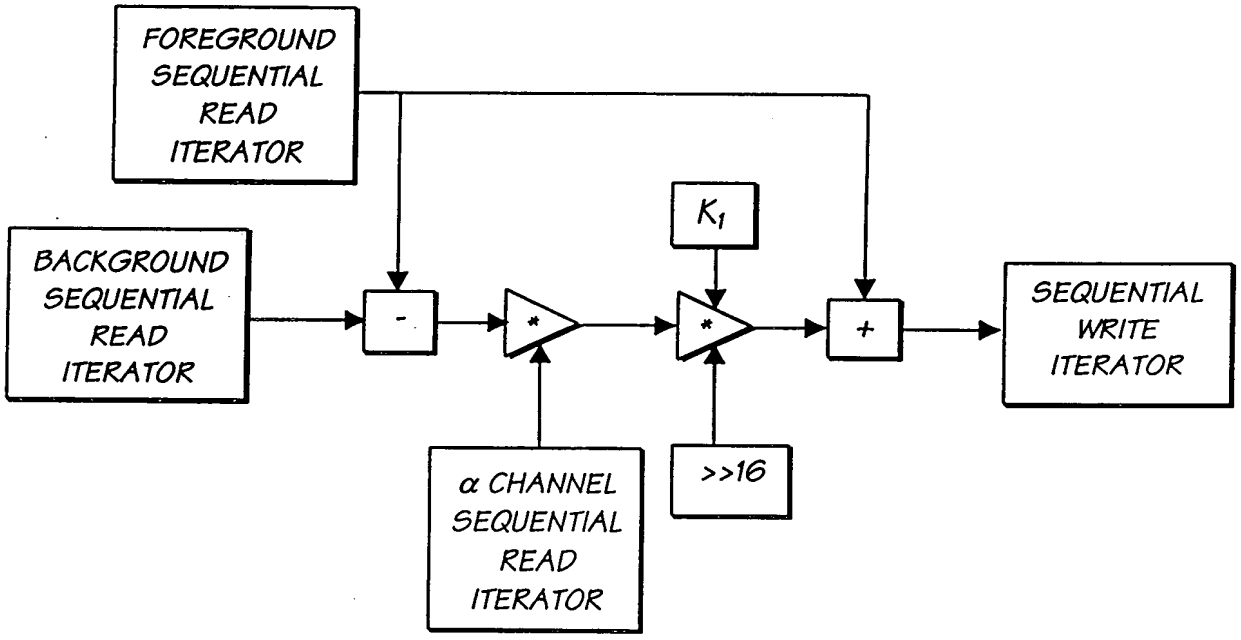

FOREGROUND SEQUENTIAL READ ITERATOR
BACKGROUND SEQUENTIAL READ ITERATOR
K1
-
*
*
+
SEQUENTIAL WRITE ITERATOR
α CHANNEL SEQUENTIAL READ ITERATOR
>>16

FIG. 84

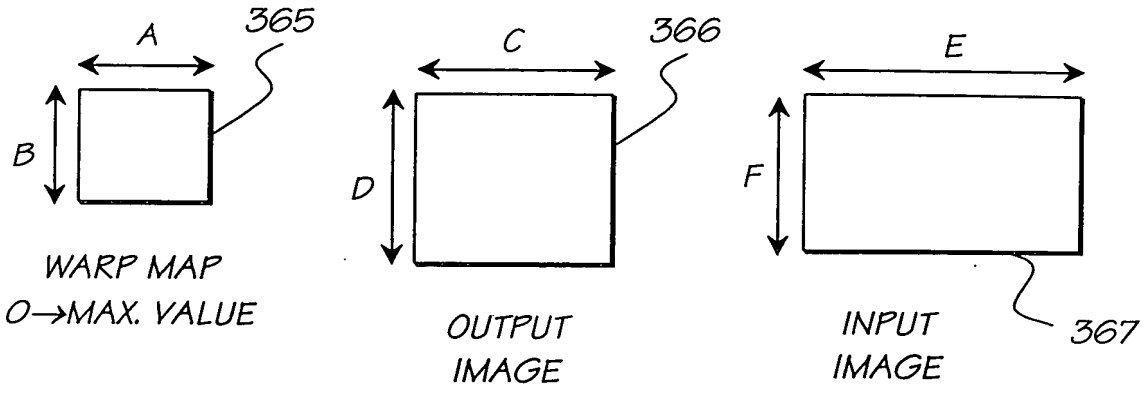


FIG. 85

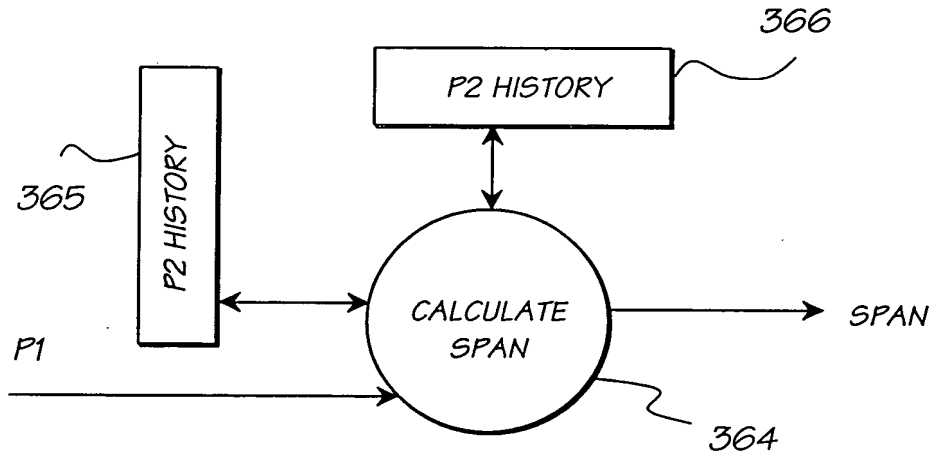


FIG. 86

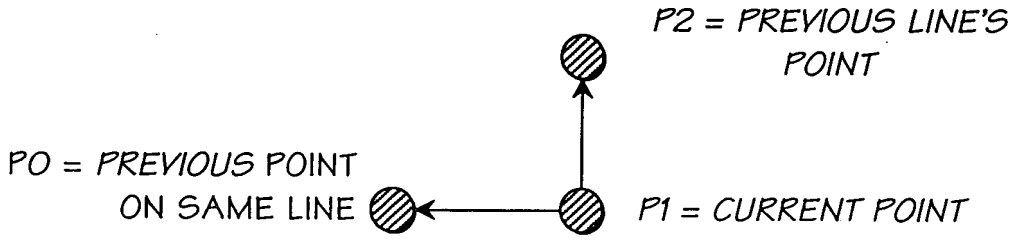


FIG. 88

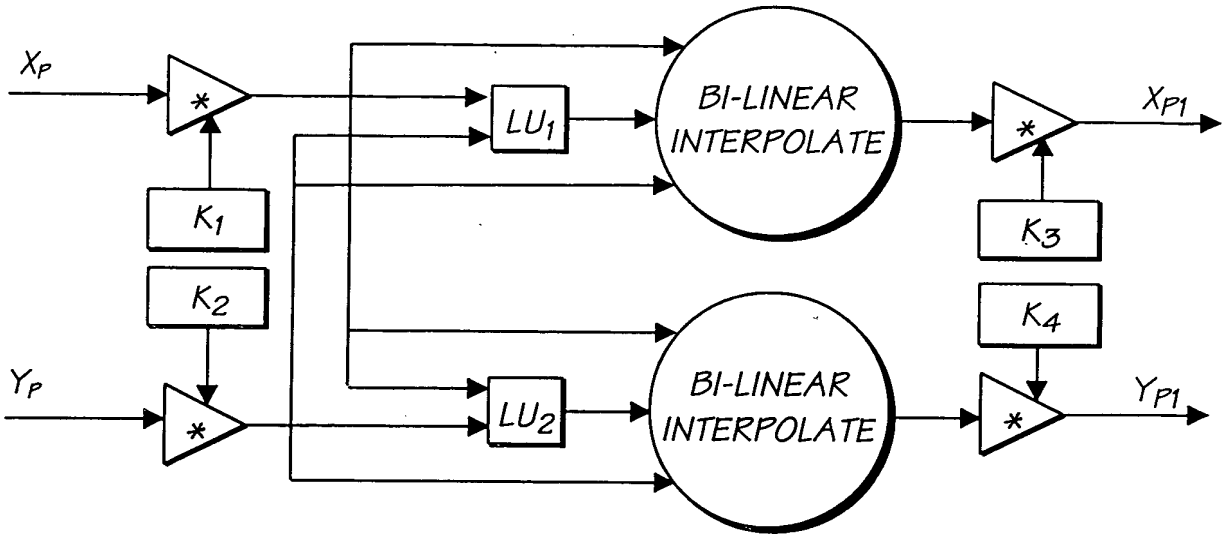

X_P
Y_P
K_1
K_2
LU_1
LU_2
BI-LINEAR INTERPOLATE
BI-LINEAR INTERPOLATE
K_3
K_4
X_{P1}
Y_{P1}

FIG. 87

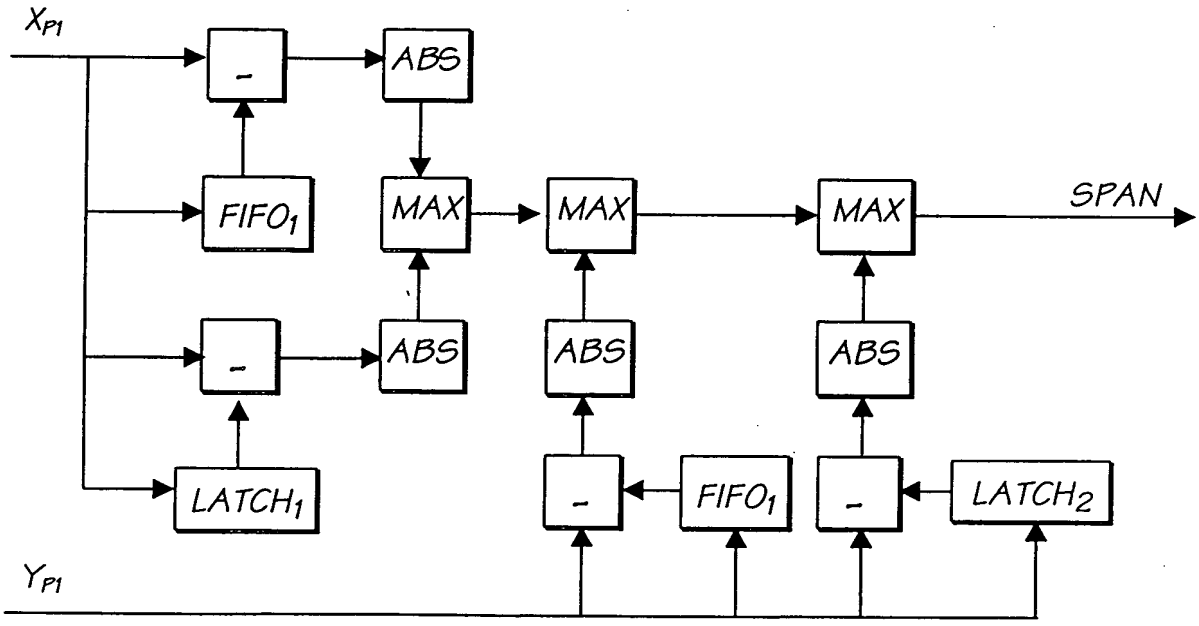

X_{P1}
Y_{P1}
ABS
ABS
ABS
$FIFO_1$
$FIFO_1$
MAX
MAX
MAX
SPAN
$LATCH_1$
$LATCH_2$

FIG. 89

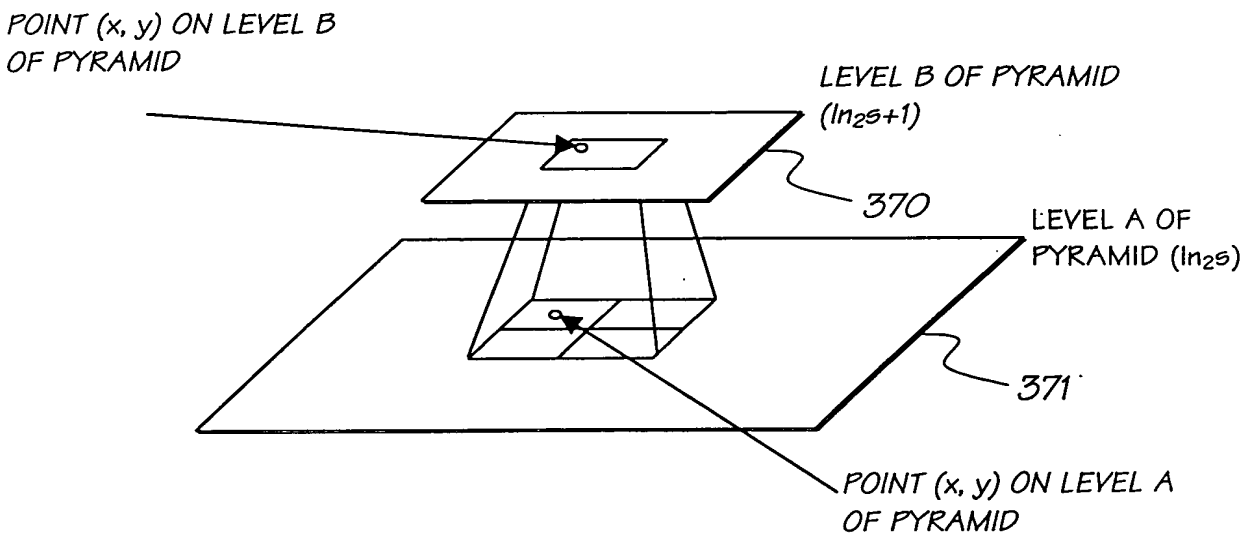

POINT (x, y) ON LEVEL B
OF PYRAMID
LEVEL B OF PYRAMID
($ln_2 s+1$)
370
LEVEL A OF
PYRAMID ($ln_2 s$)
371
POINT (x, y) ON LEVEL A
OF PYRAMID

FIG. 90

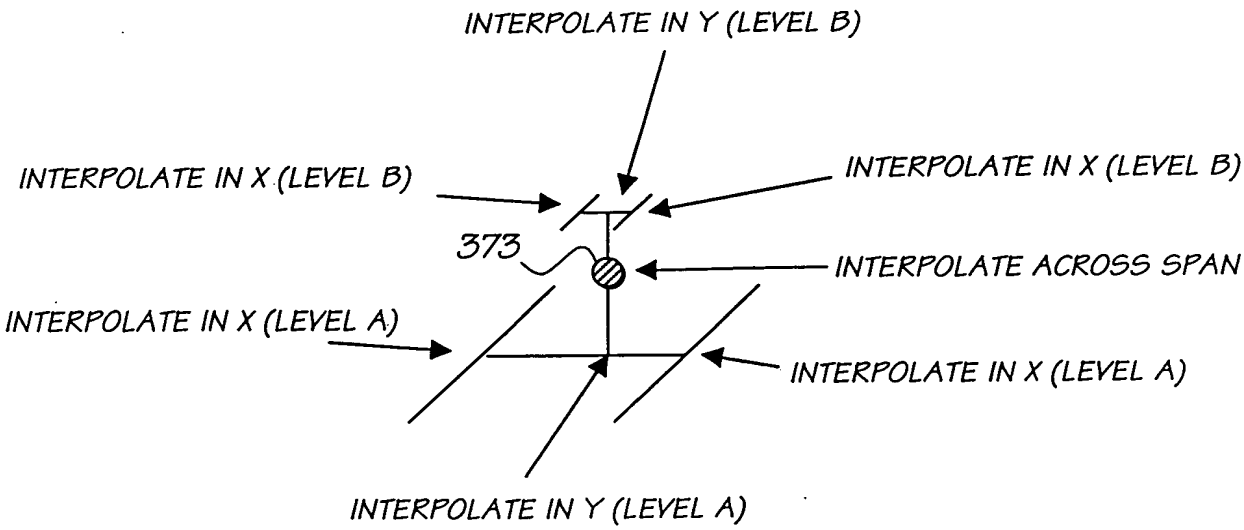

INTERPOLATE IN Y (LEVEL B)
INTERPOLATE IN X (LEVEL B)
INTERPOLATE IN X (LEVEL B)
373
INTERPOLATE ACROSS SPAN
INTERPOLATE IN X (LEVEL A)
INTERPOLATE IN X (LEVEL A)
INTERPOLATE IN Y (LEVEL A)

FIG. 91

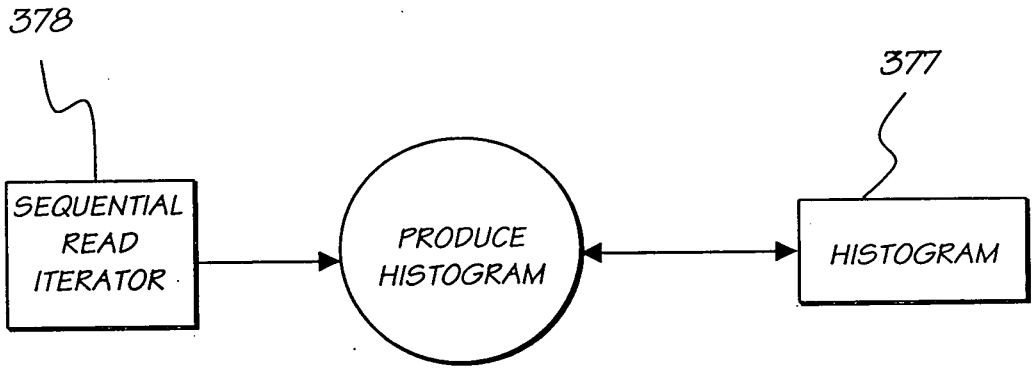

378
377
SEQUENTIAL READ ITERATOR
PRODUCE HISTOGRAM
HISTOGRAM

FIG. 92

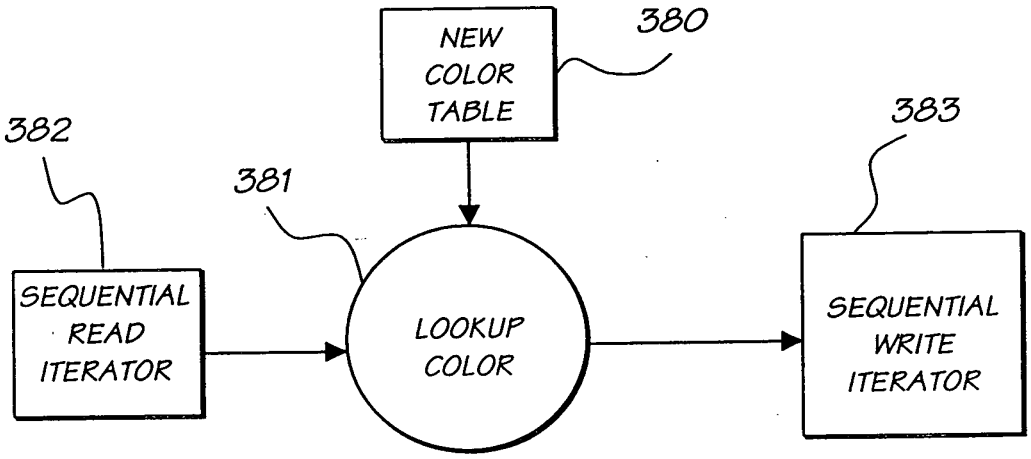

NEW COLOR TABLE
380
382
381
383
SEQUENTIAL READ ITERATOR
LOOKUP COLOR
SEQUENTIAL WRITE ITERATOR

FIG. 93

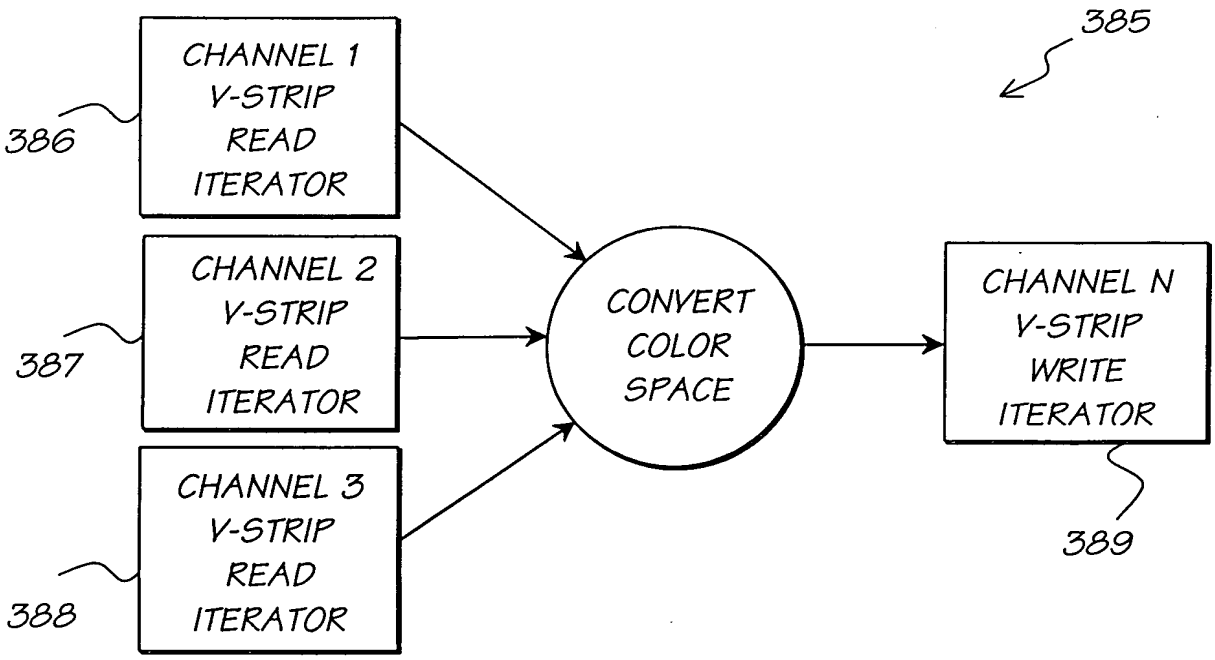


FIG. 94

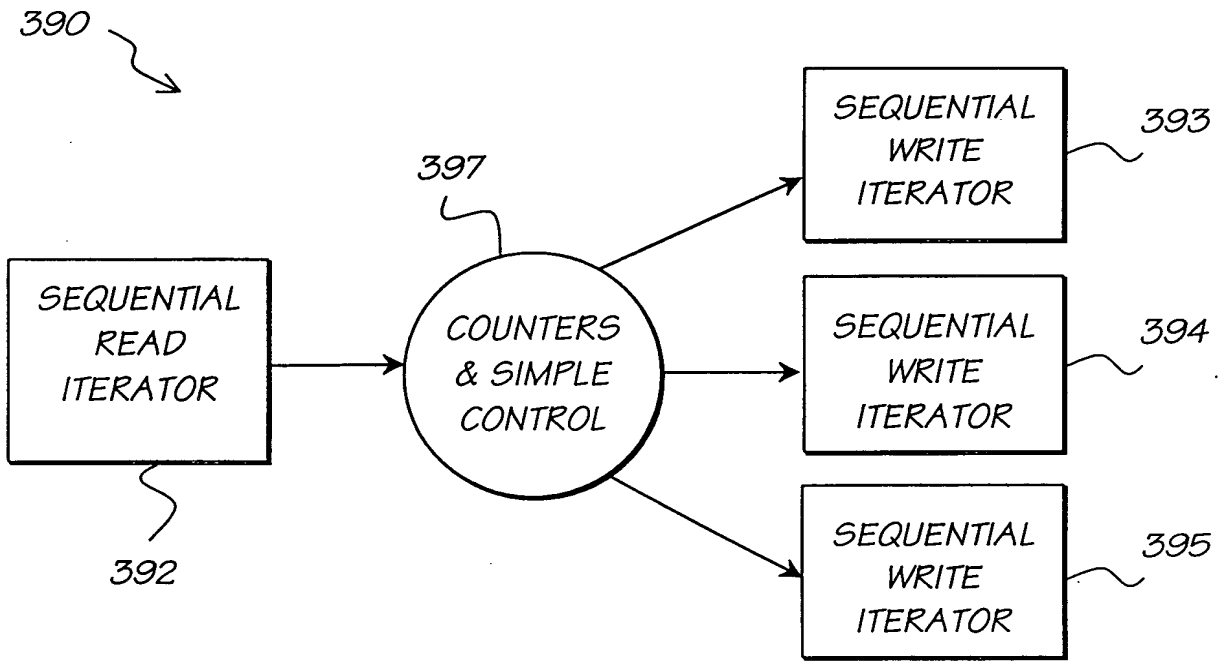


FIG. 101

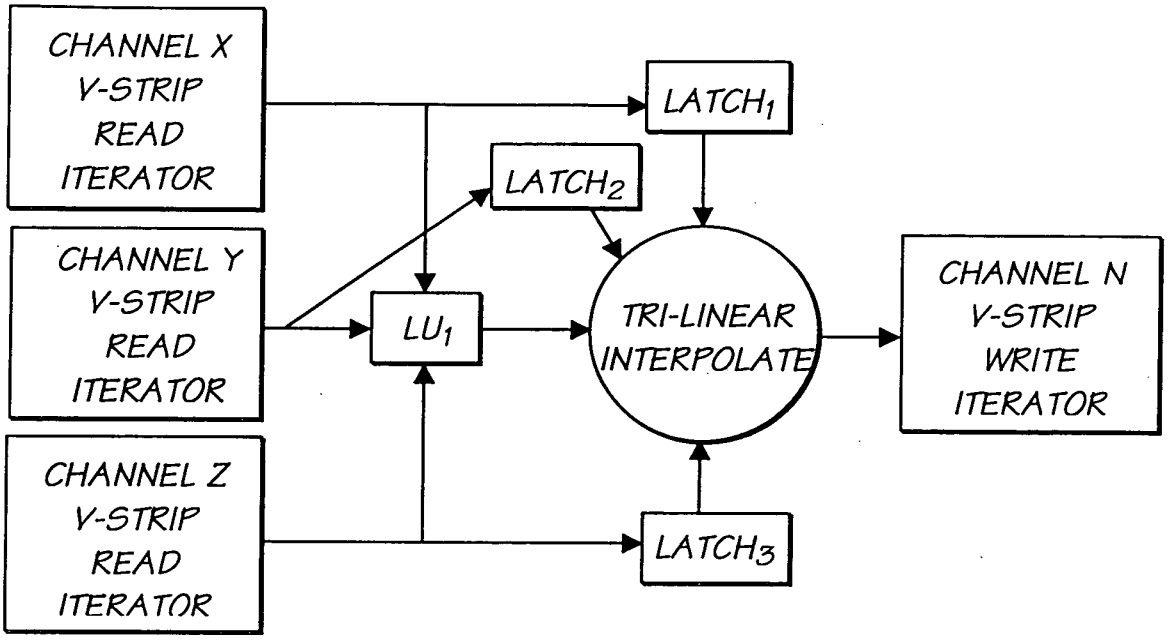

CHANNEL X V-STRIP READ ITERATOR
CHANNEL Y V-STRIP READ ITERATOR
CHANNEL Z V-STRIP READ ITERATOR
LATCH1
LATCH2
LATCH3
LU1
TRI-LINEAR INTERPOLATE
CHANNEL N V-STRIP WRITE ITERATOR

FIG. 95

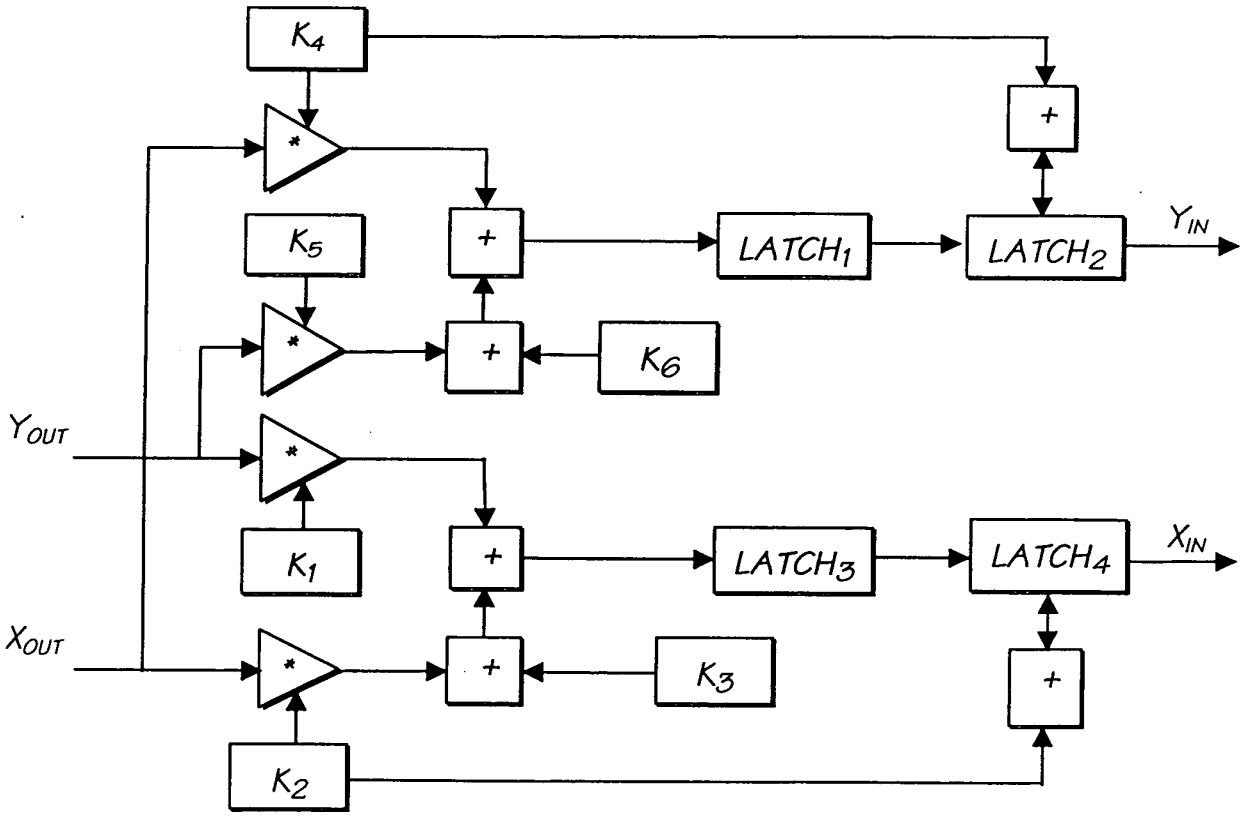

K4
K5
K6
K1
K2
K3
YOUT
XOUT
LATCH1
LATCH2
LATCH3
LATCH4
YIN
XIN
+
*

FIG. 96

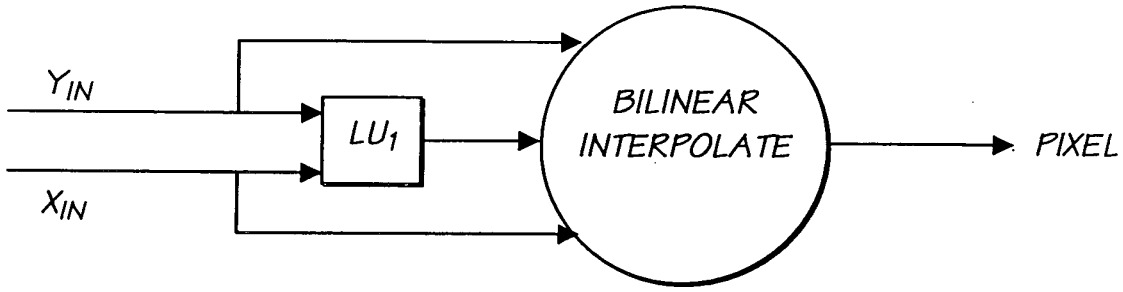

Y_{IN}
X_{IN}
LU_1
BILINEAR INTERPOLATE
PIXEL

FIG. 97

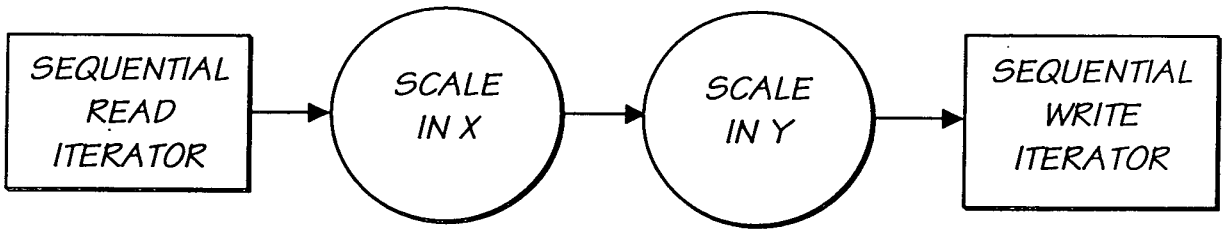

SEQUENTIAL READ ITERATOR
SCALE IN X
SCALE IN Y
SEQUENTIAL WRITE ITERATOR

FIG. 98

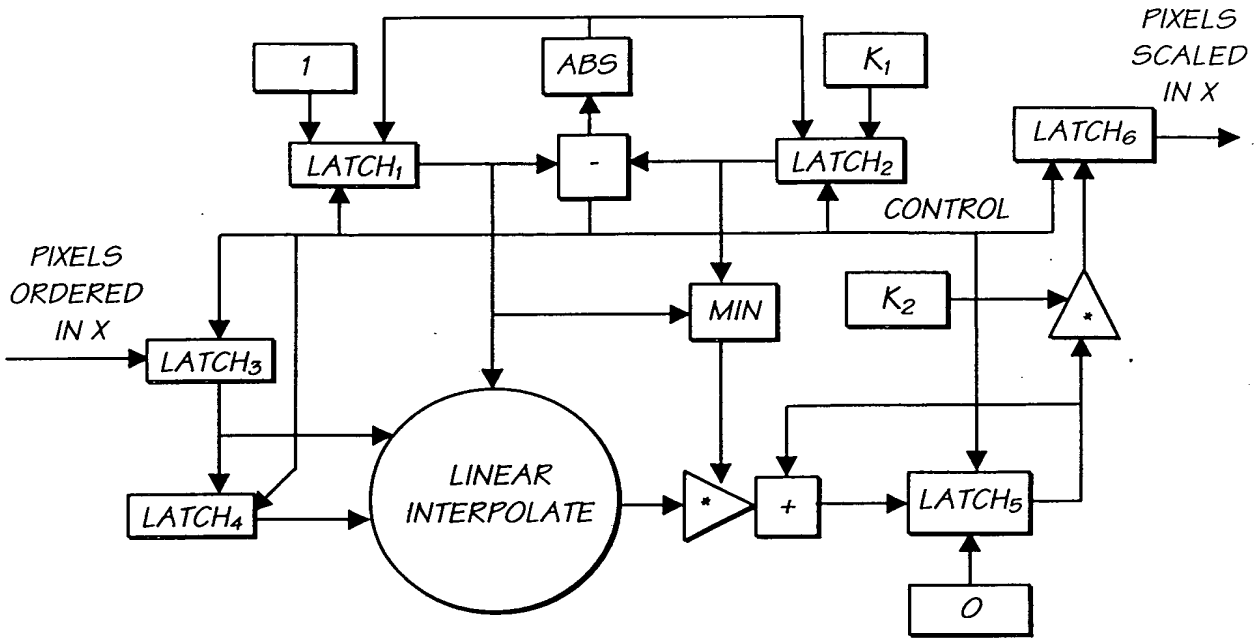


FIG. 99

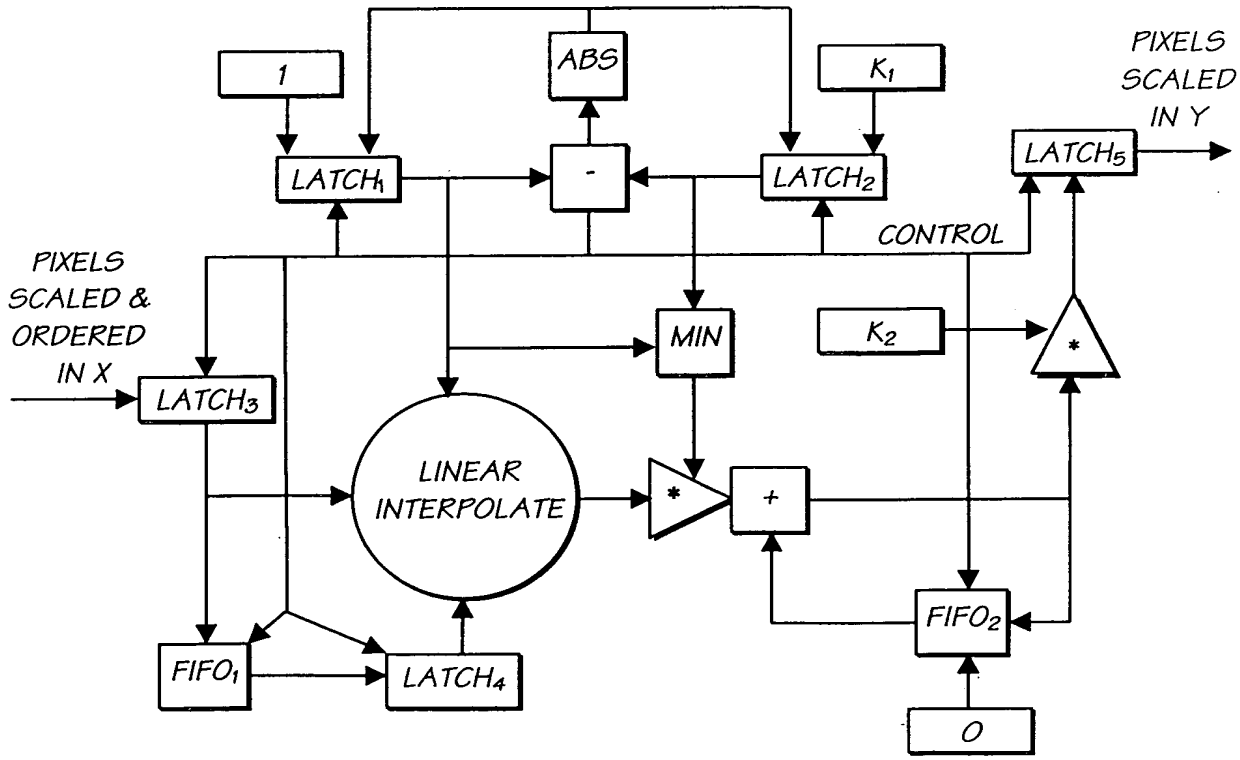


FIG. 100

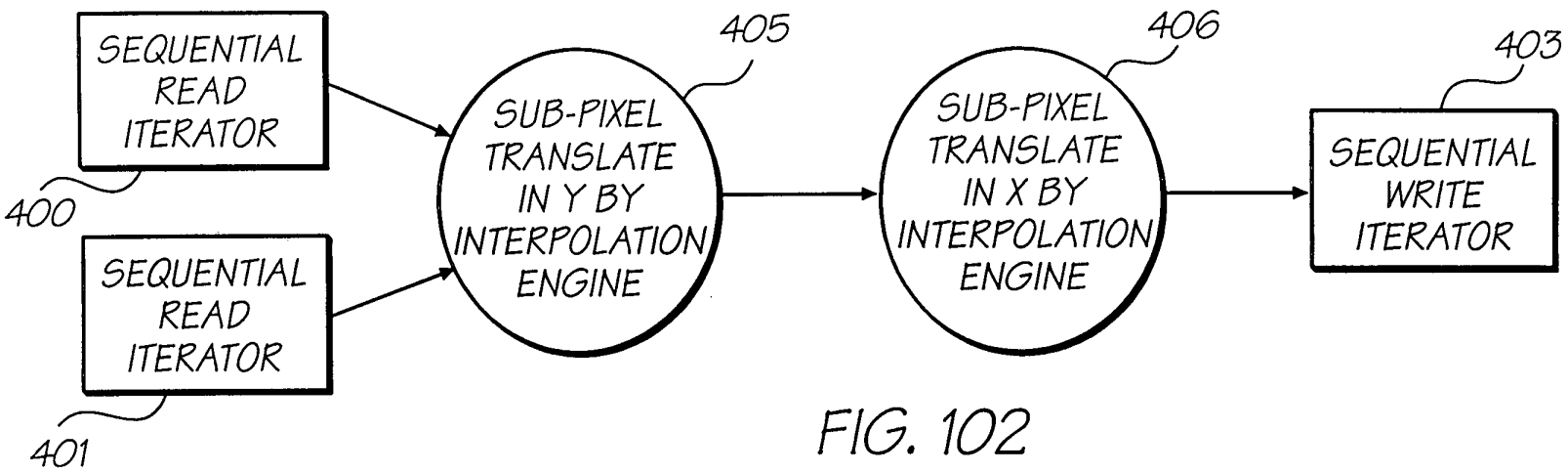


FIG. 102

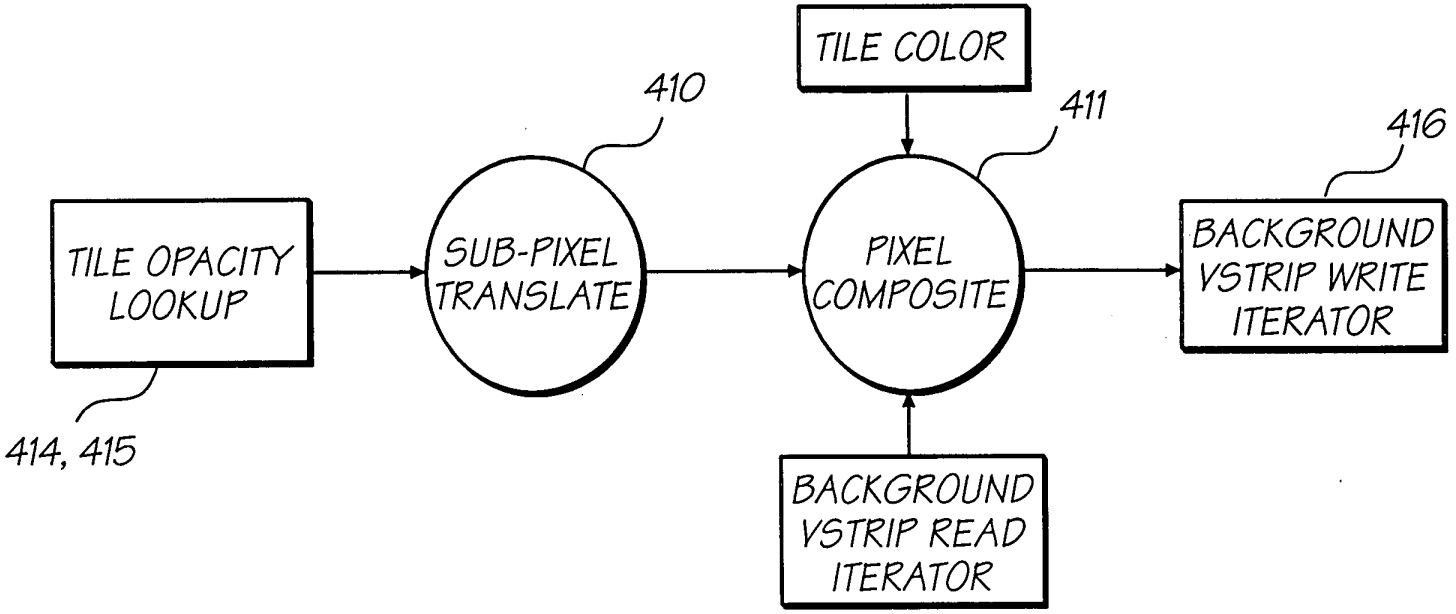


FIG. 103

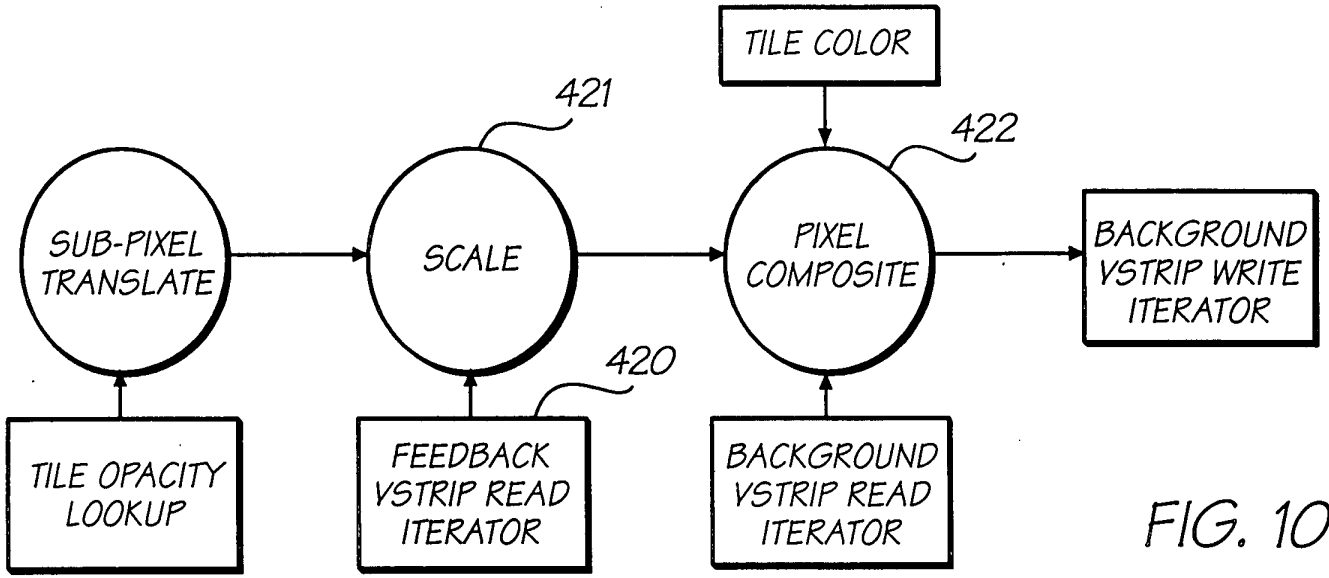


FIG. 104

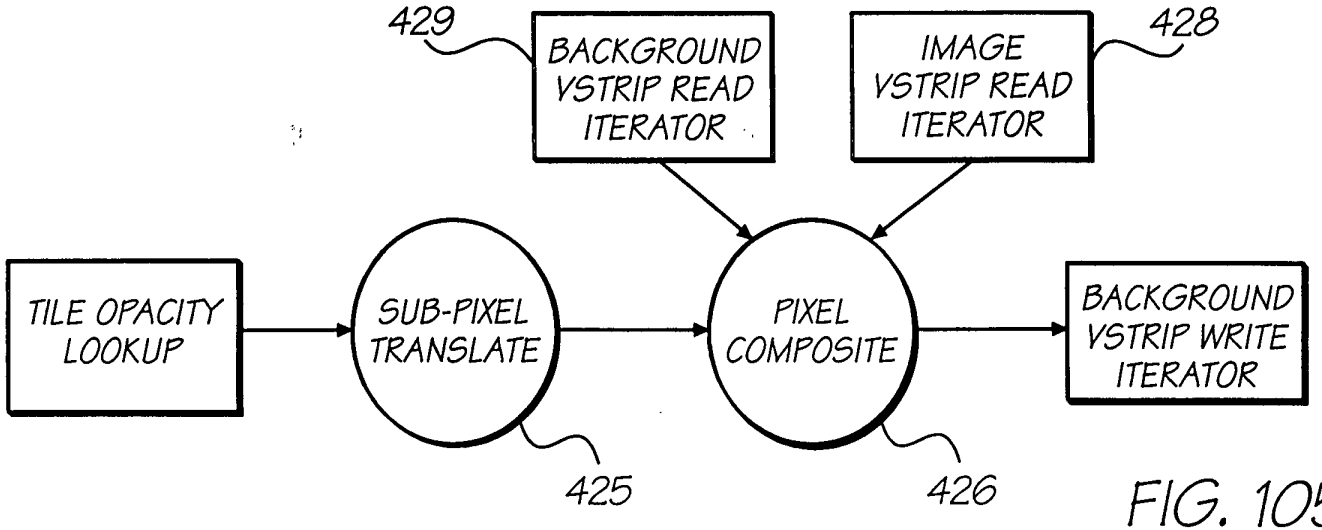


FIG. 105

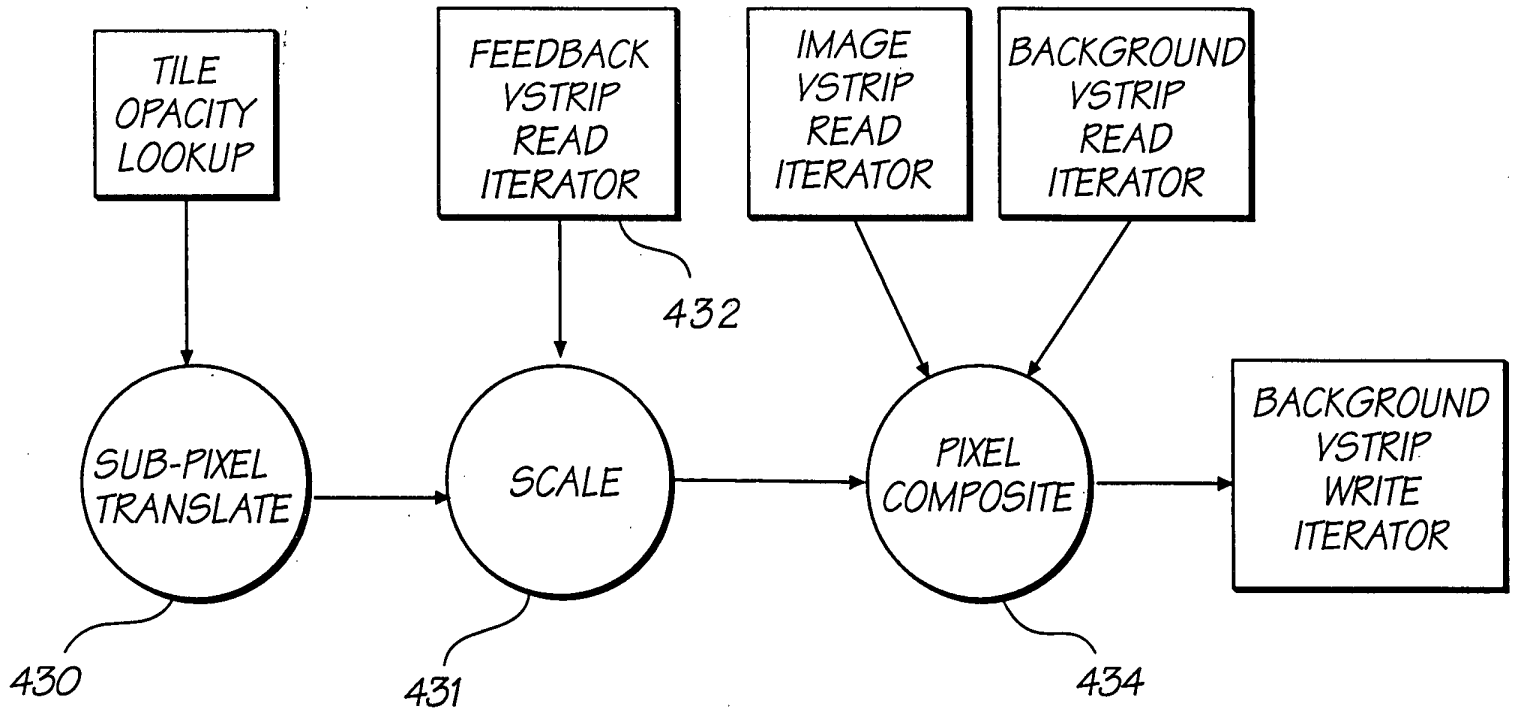


FIG. 106

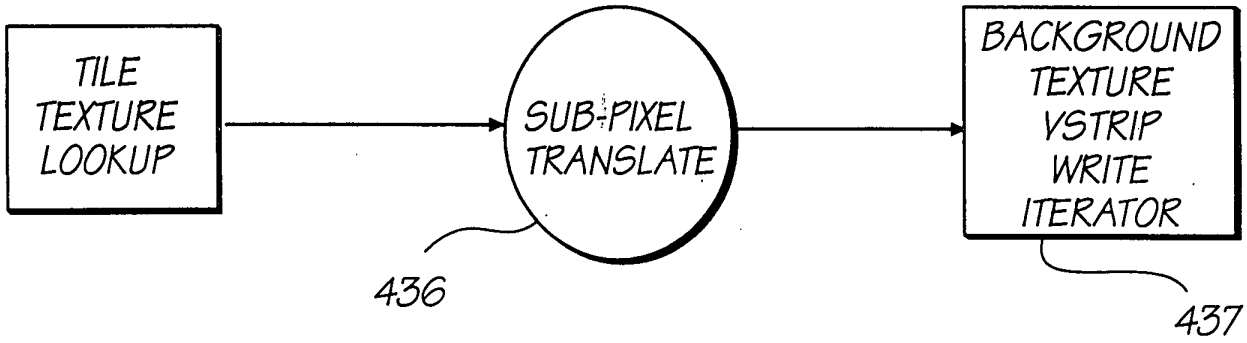


FIG. 107

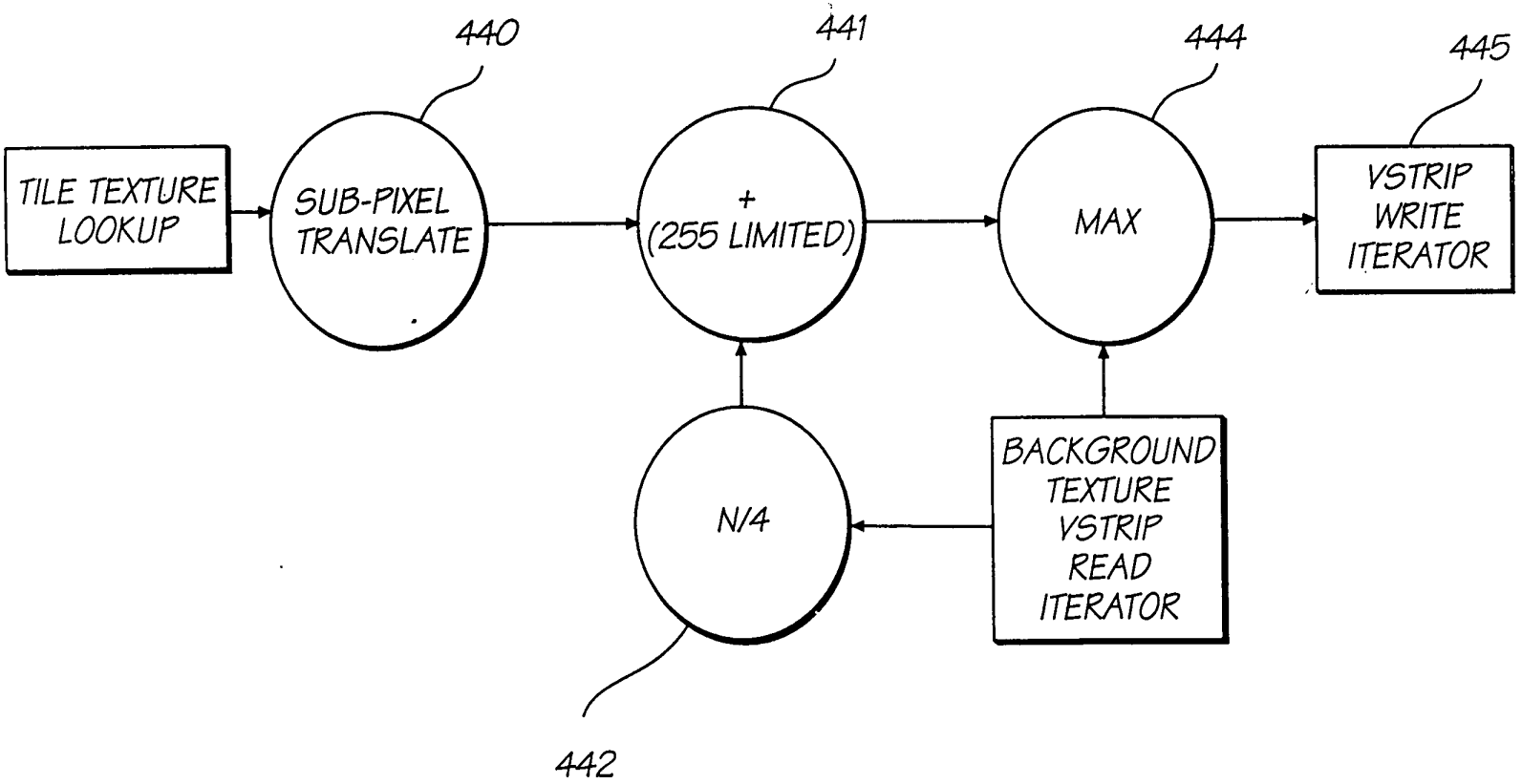

440
441
444
445
TILE TEXTURE LOOKUP
SUB-PIXEL TRANSLATE
+ (255 LIMITED)
MAX
VSTRIP WRITE ITERATOR
N/4
BACKGROUND TEXTURE VSTRIP READ ITERATOR
442

FIG. 108

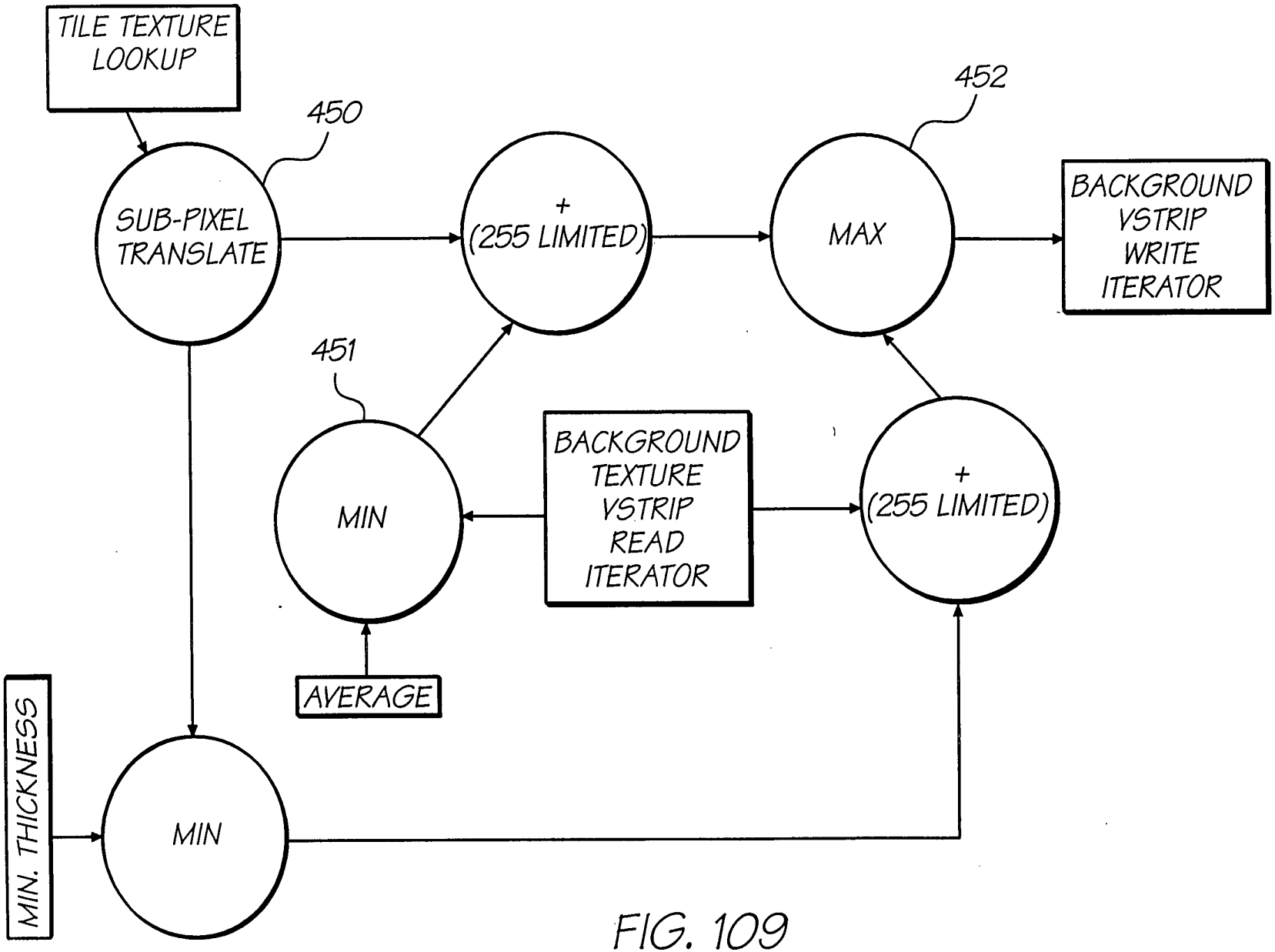

TILE TEXTURE LOOKUP
450
SUB-PIXEL TRANSLATE
+ (255 LIMITED)
452
MAX
BACKGROUND VSTRIP WRITE ITERATOR
451
MIN
BACKGROUND TEXTURE VSTRIP READ ITERATOR
+ (255 LIMITED)
AVERAGE
MIN. THICKNESS
MIN

FIG. 109

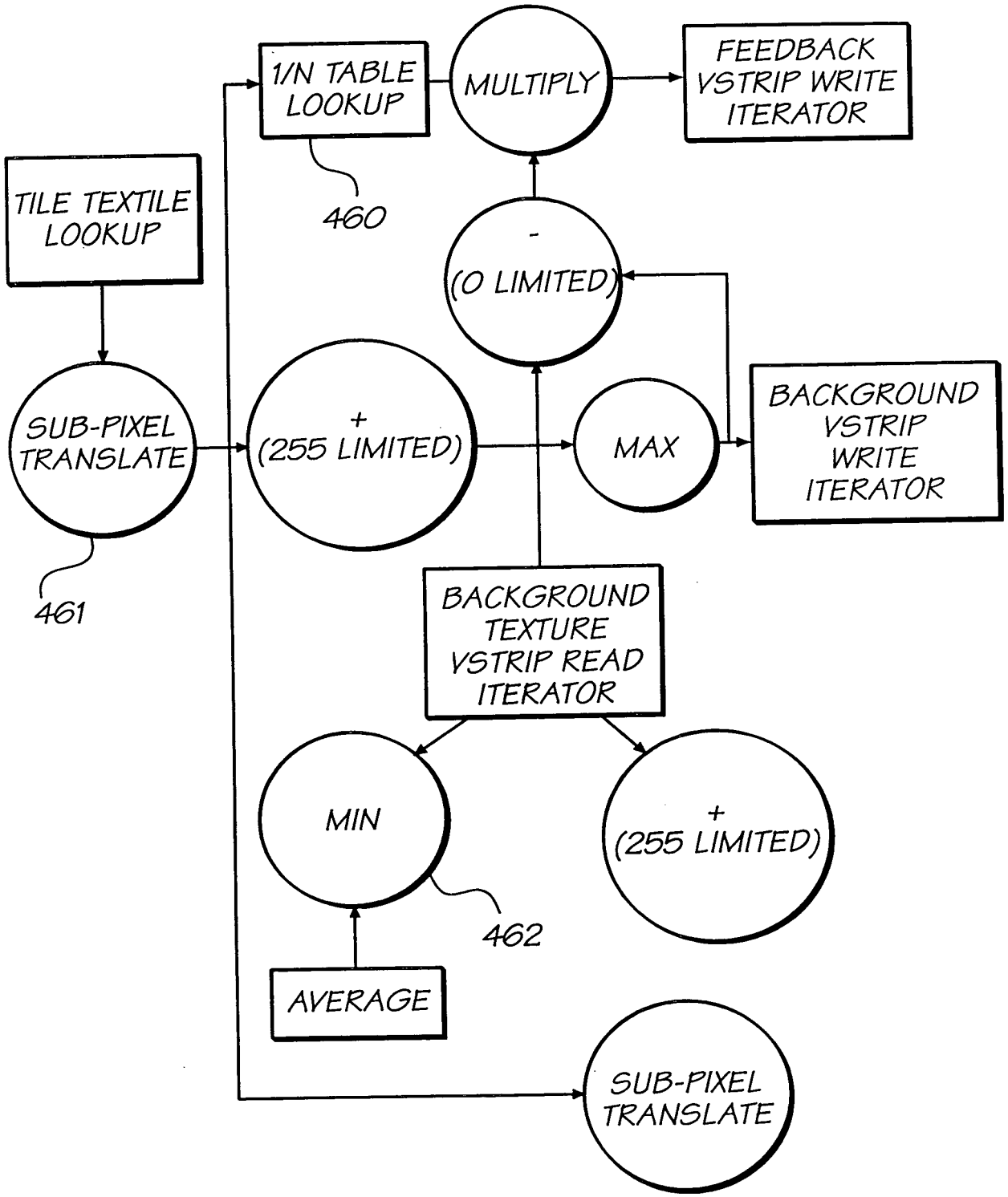

1/N TABLE LOOKUP
MULTIPLY
FEEDBACK VSTRIP WRITE ITERATOR
460
TILE TEXTILE LOOKUP
-
(0 LIMITED)
SUB-PIXEL TRANSLATE
+
(255 LIMITED)
MAX
BACKGROUND VSTRIP WRITE ITERATOR
461
BACKGROUND TEXTURE VSTRIP READ ITERATOR
MIN
+
(255 LIMITED)
462
AVERAGE
SUB-PIXEL TRANSLATE

FIG. 110

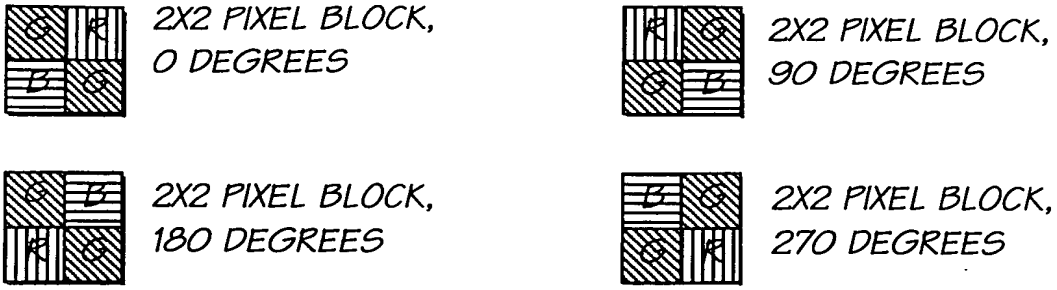


FIG. 111

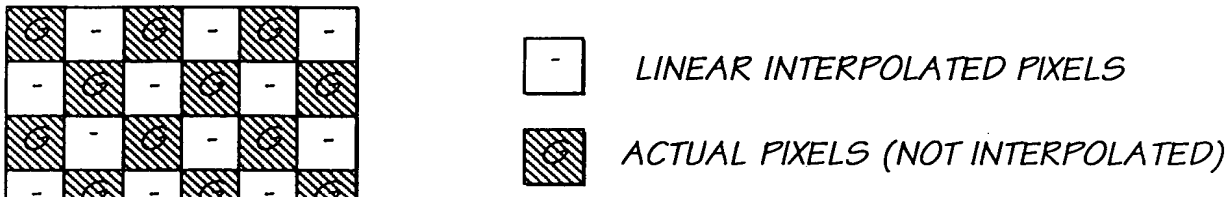


FIG. 112

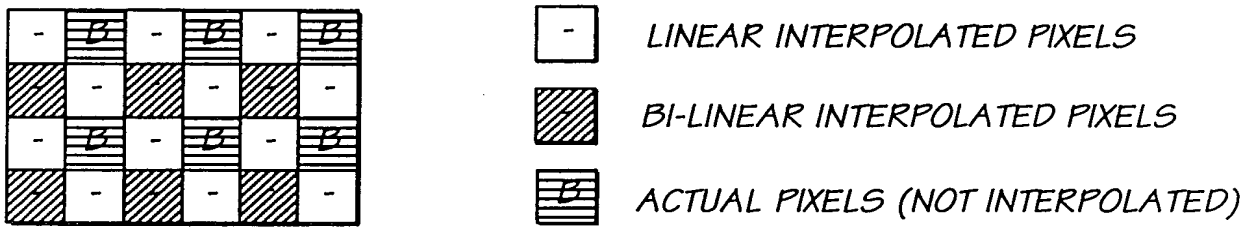


FIG. 113

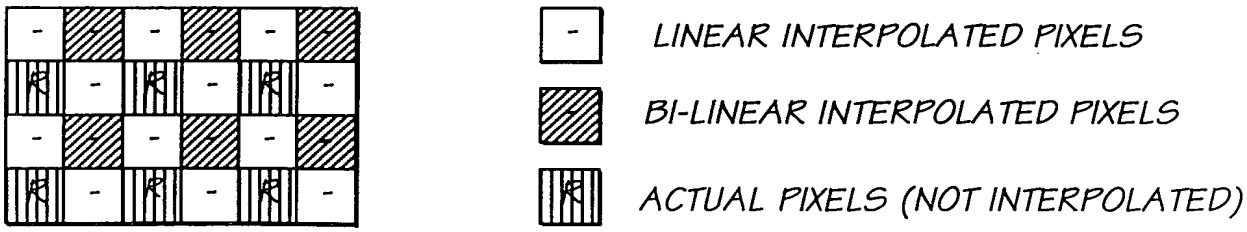


FIG. 114

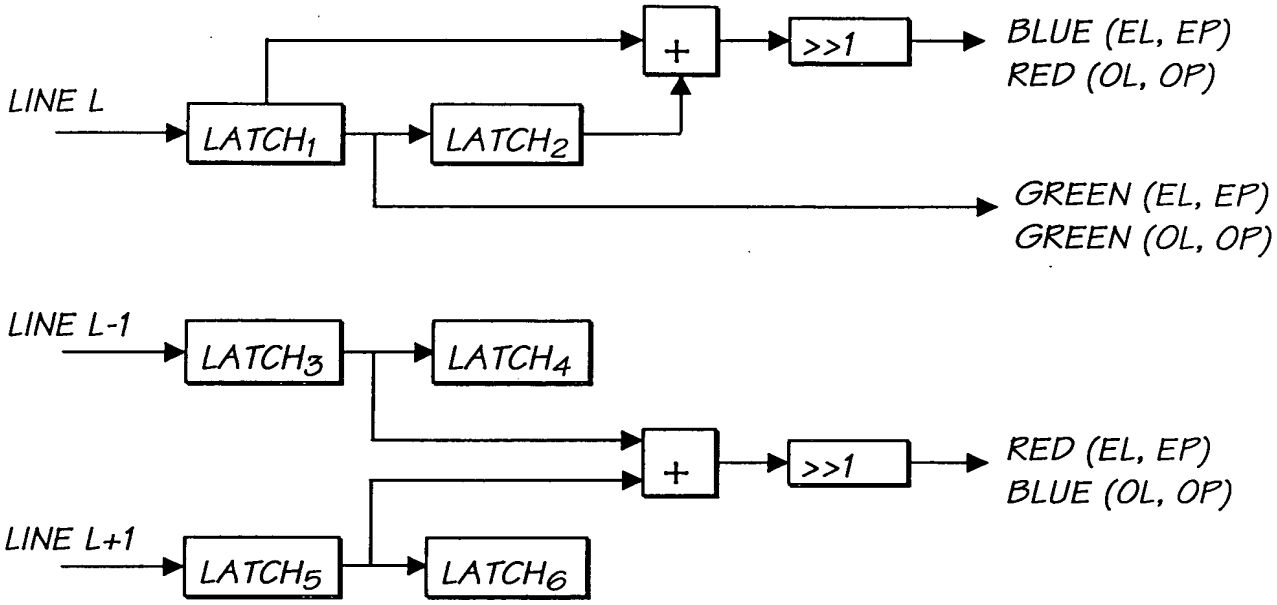


FIG. 115

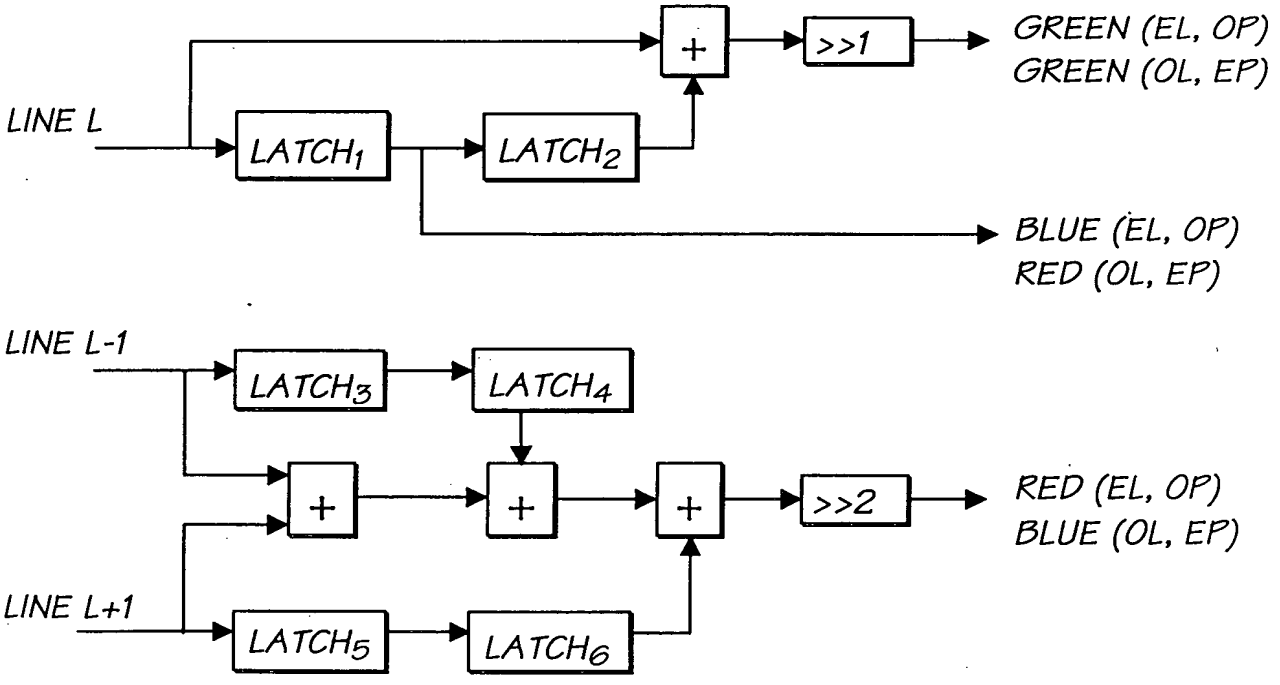


FIG. 116

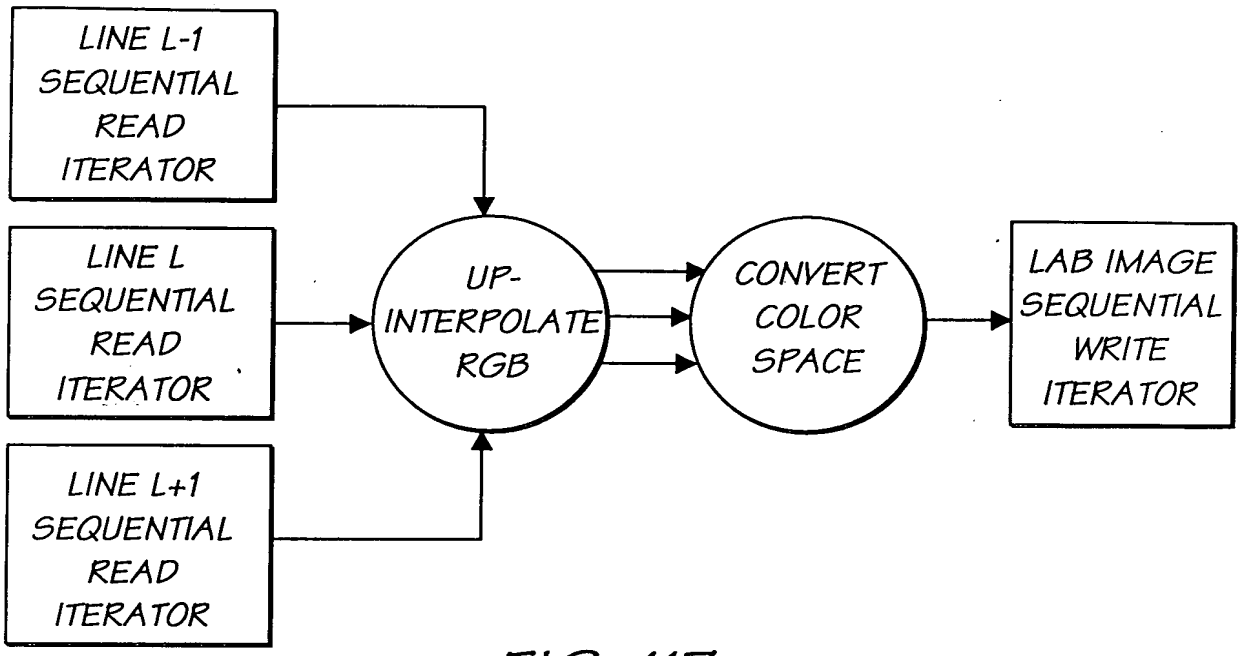


FIG. 117

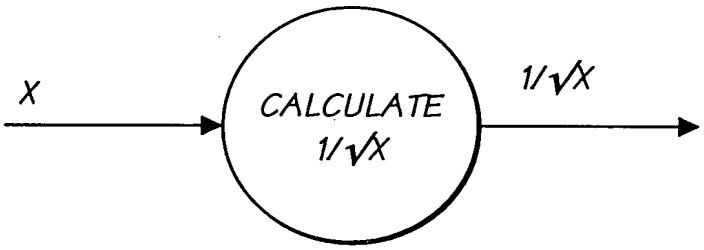


FIG. 118

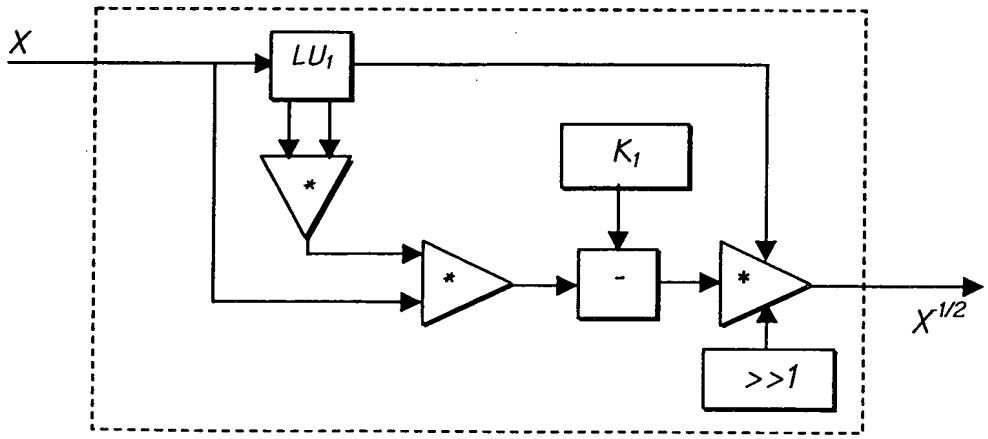


FIG. 119

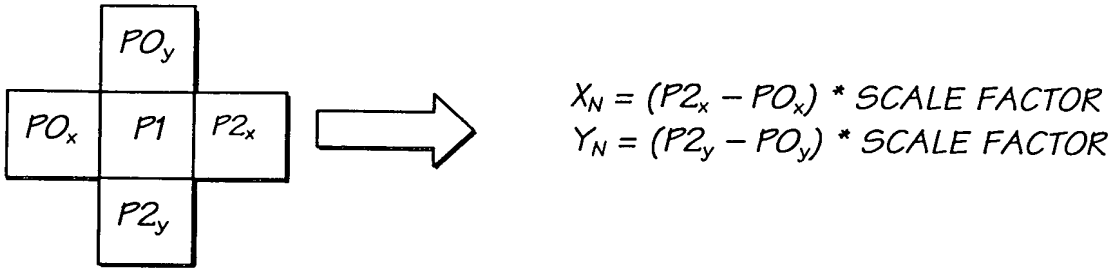


FIG. 120

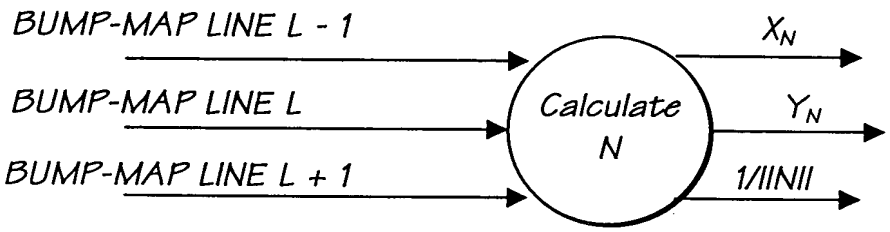


FIG. 121

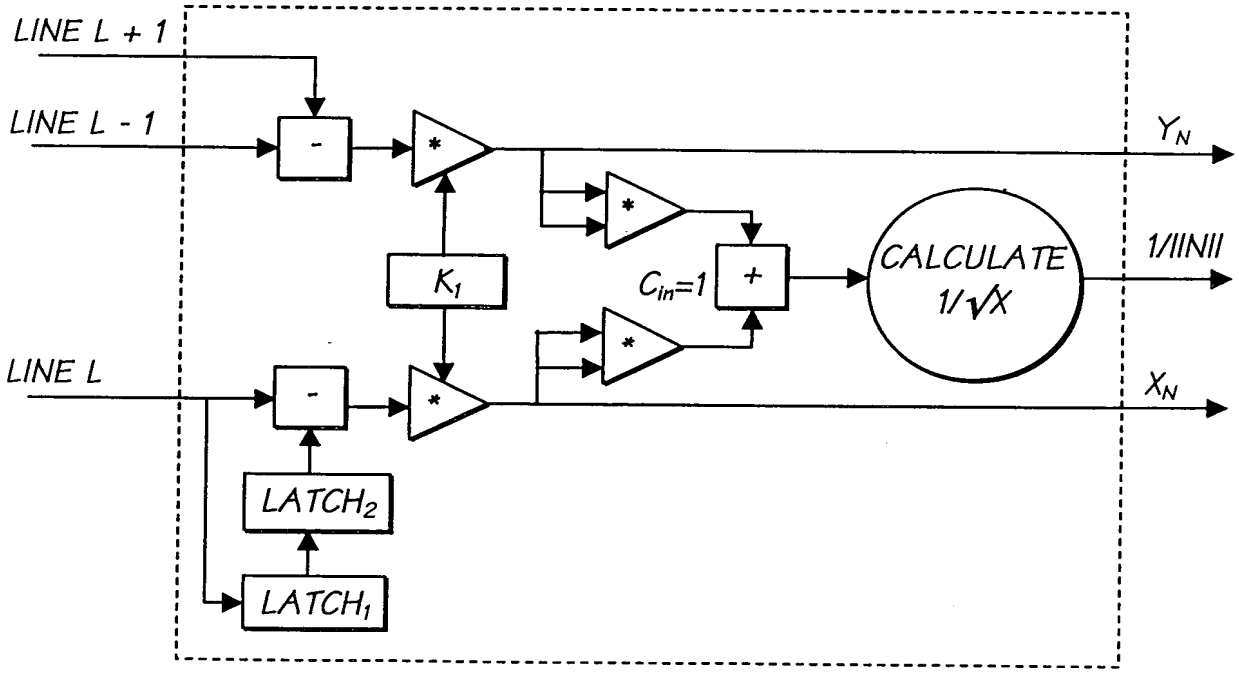


FIG. 122

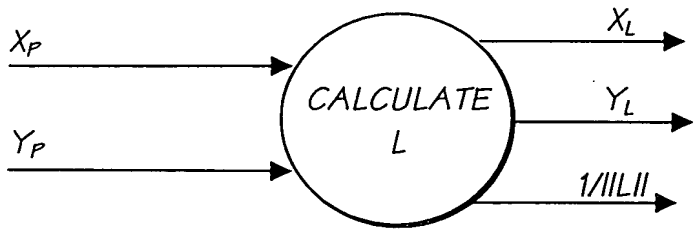


FIG. 123

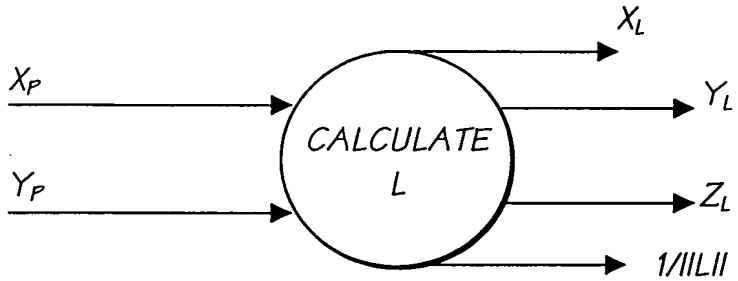


FIG. 124

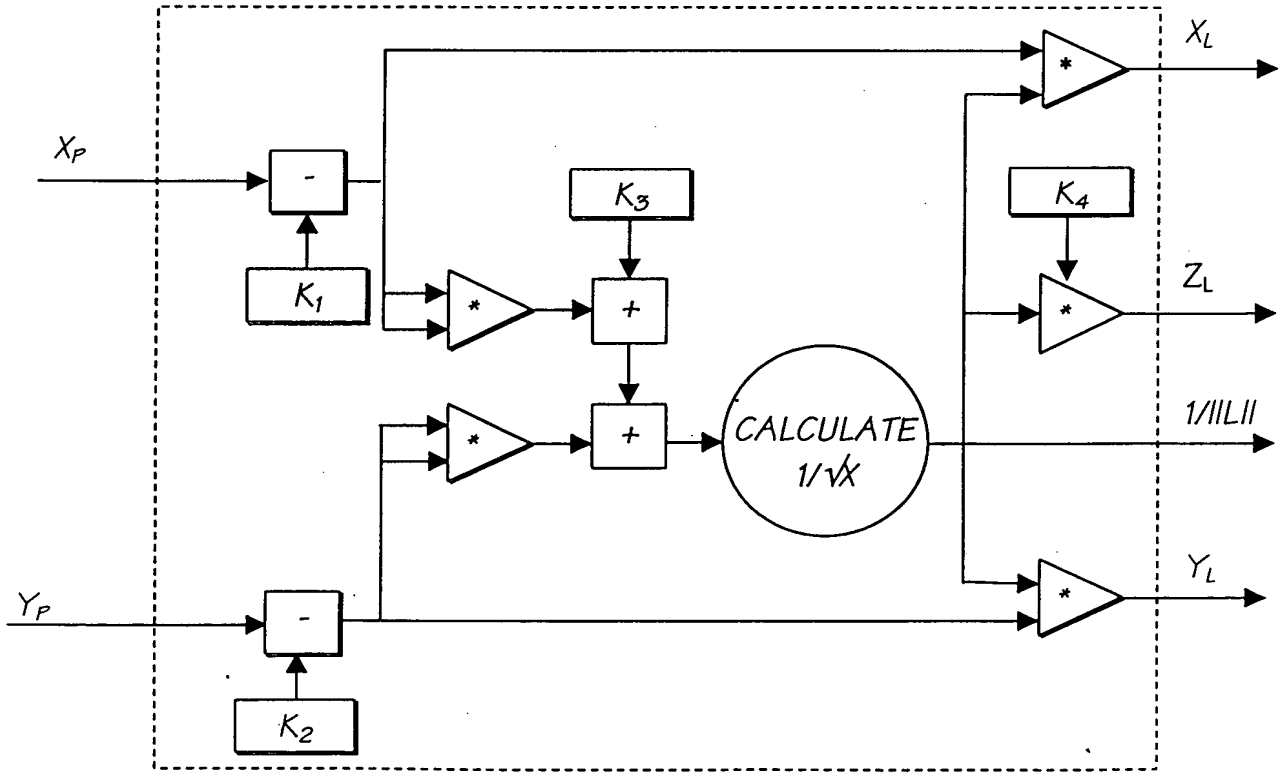


FIG. 125

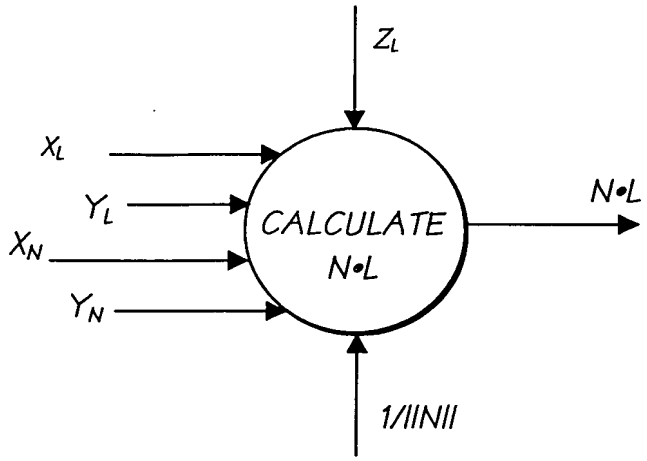


FIG. 126

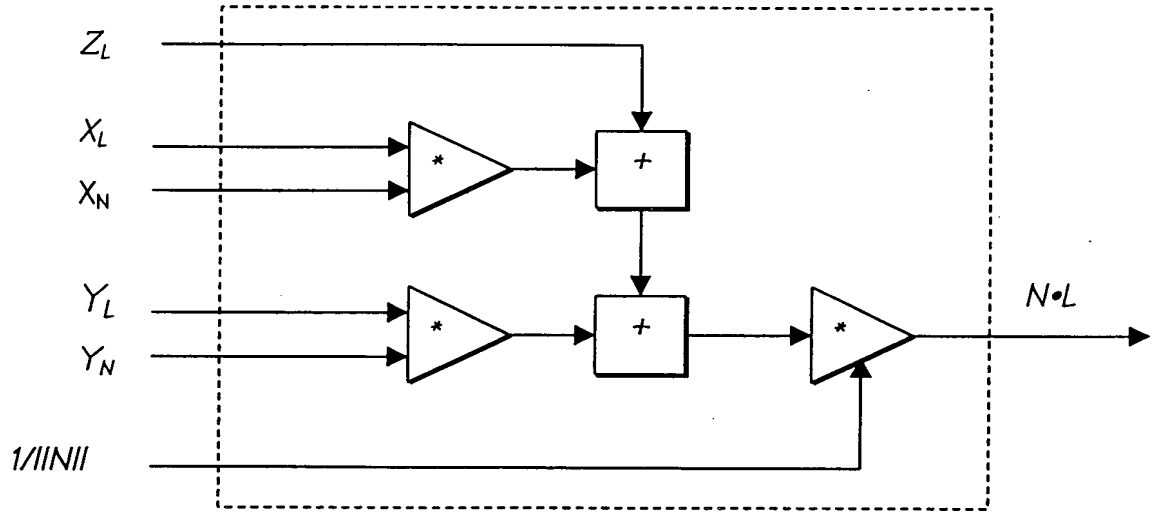


FIG. 127

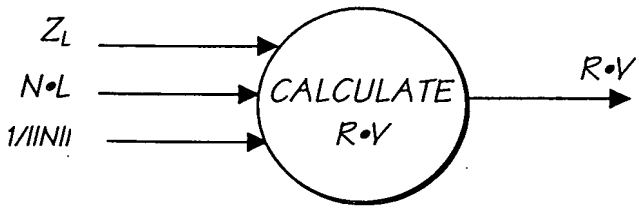


FIG. 128

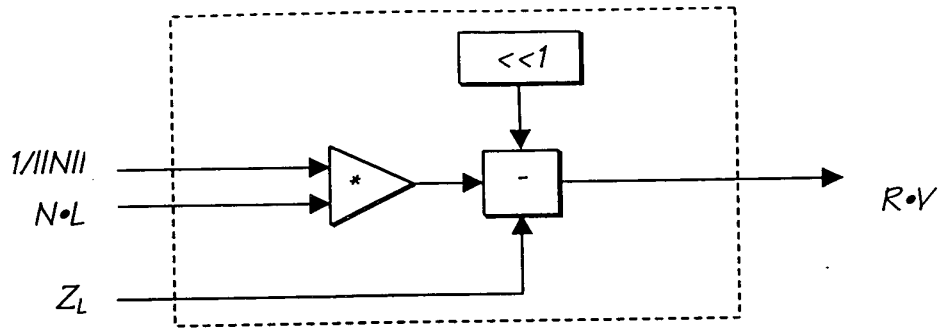


FIG. 129

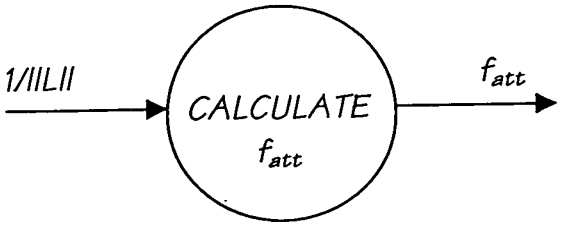


FIG. 130

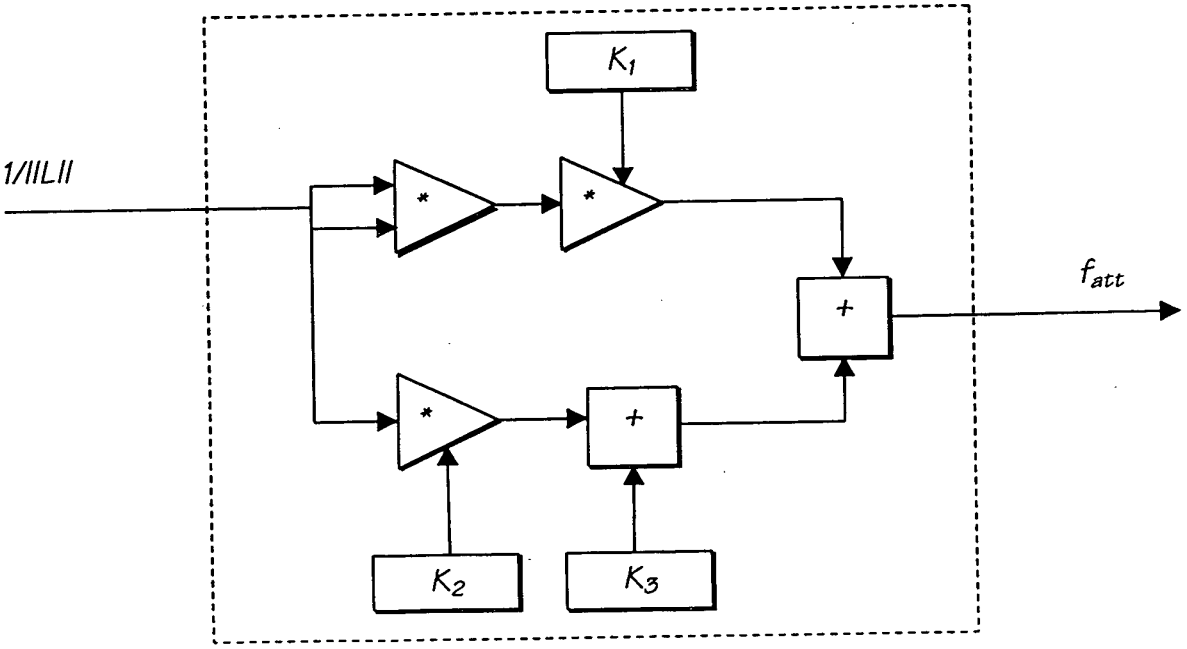


FIG. 131

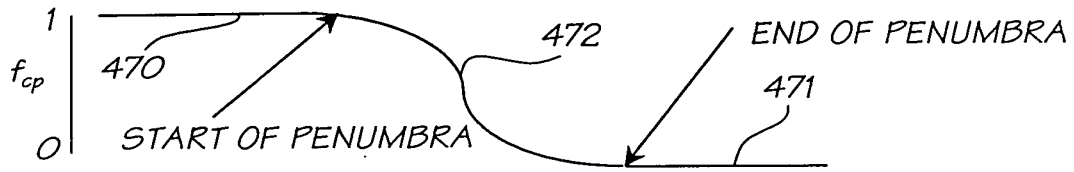


FIG. 132

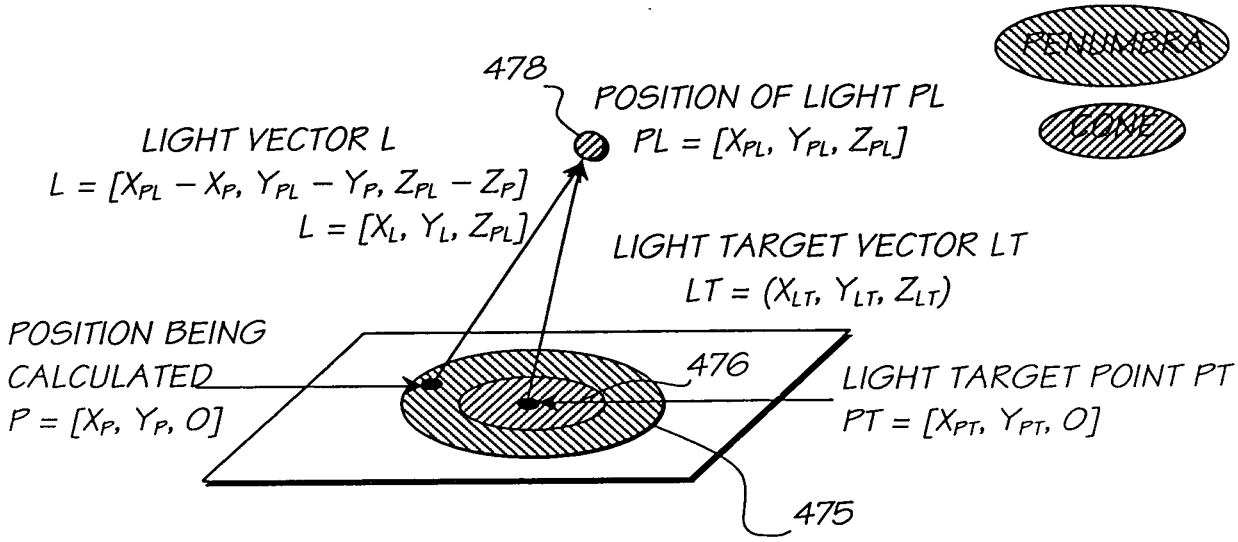


FIG. 133

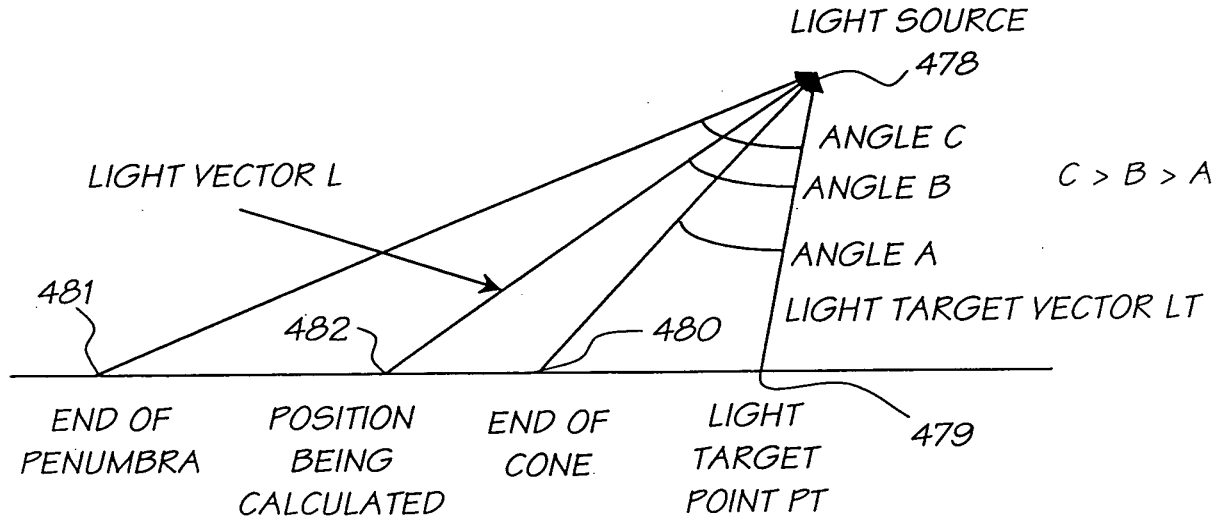


FIG. 134

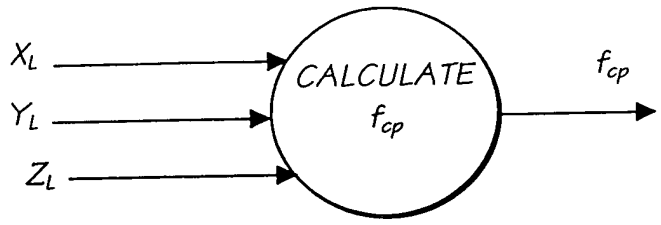


FIG. 135

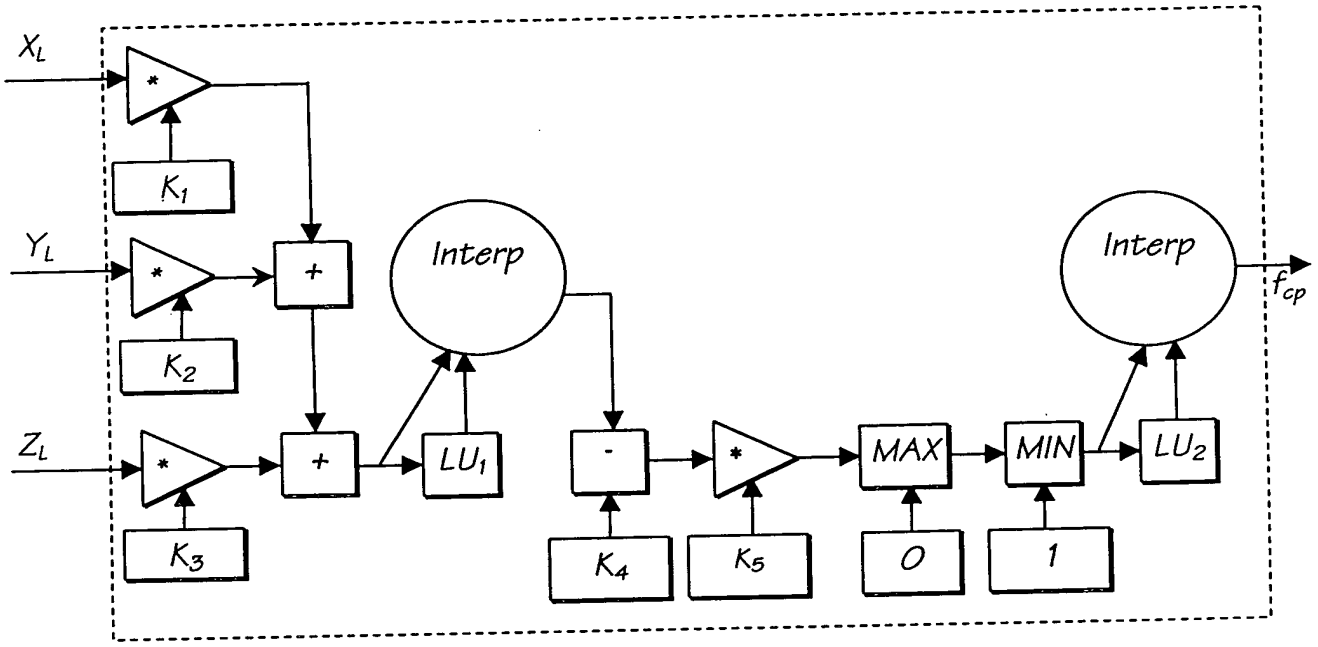


FIG. 136

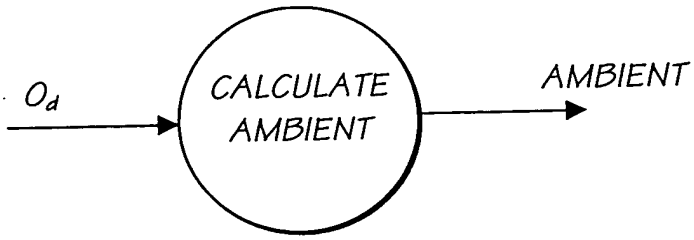


FIG. 137

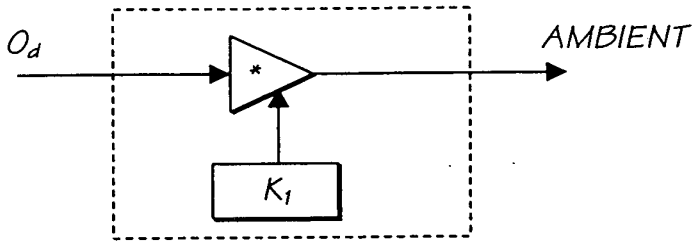


FIG. 138

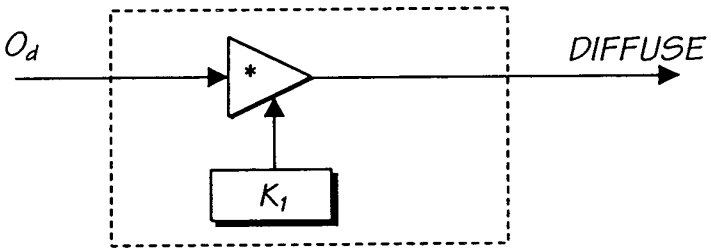


FIG. 139

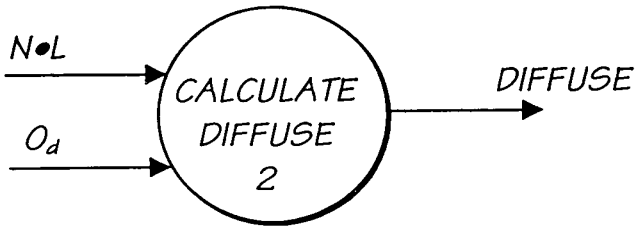


FIG. 140

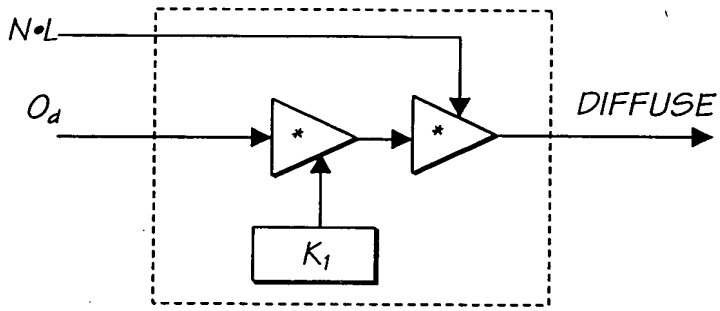


FIG. 141

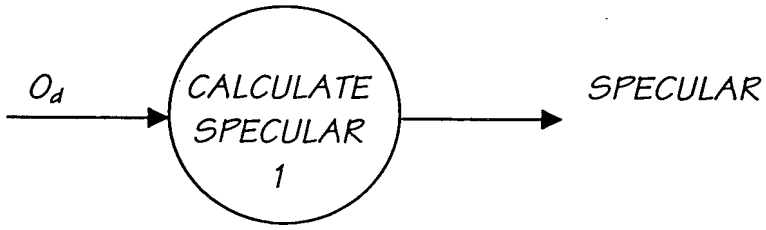


FIG. 142

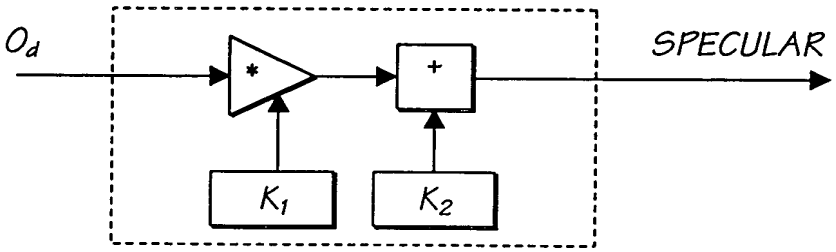


FIG. 143

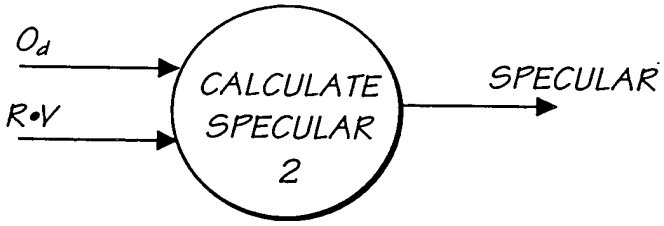


FIG. 144

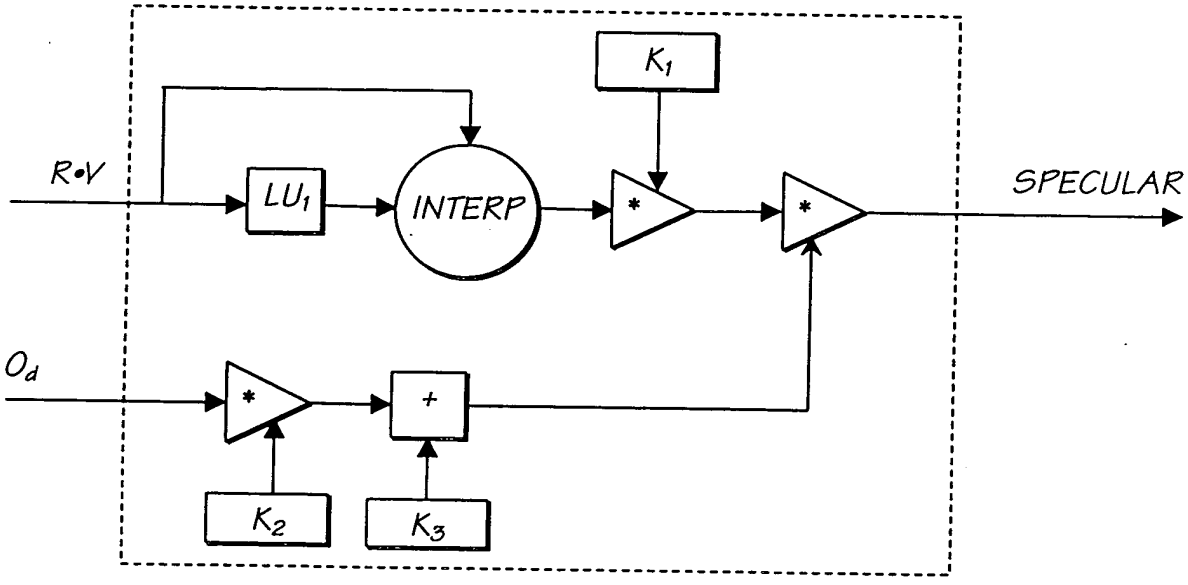


FIG. 145

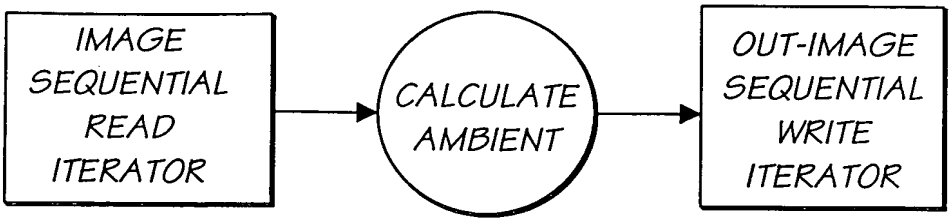


FIG. 146

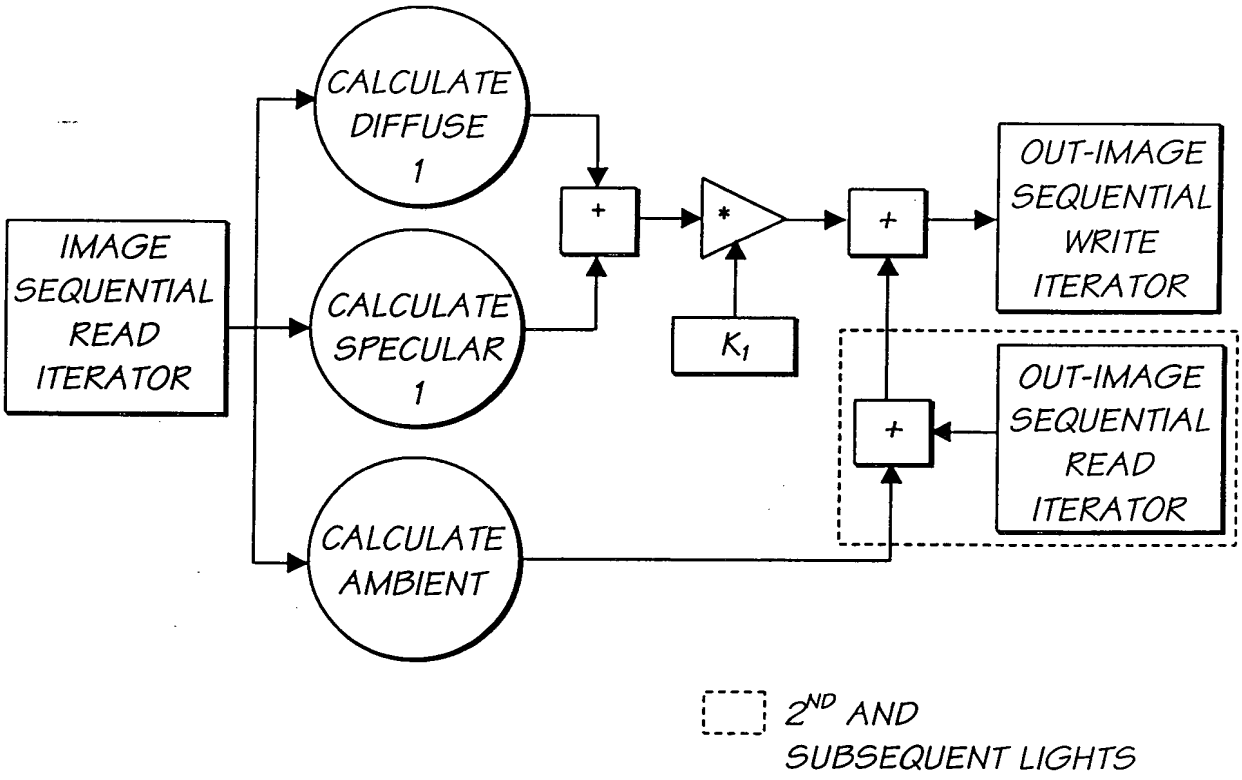


FIG. 147

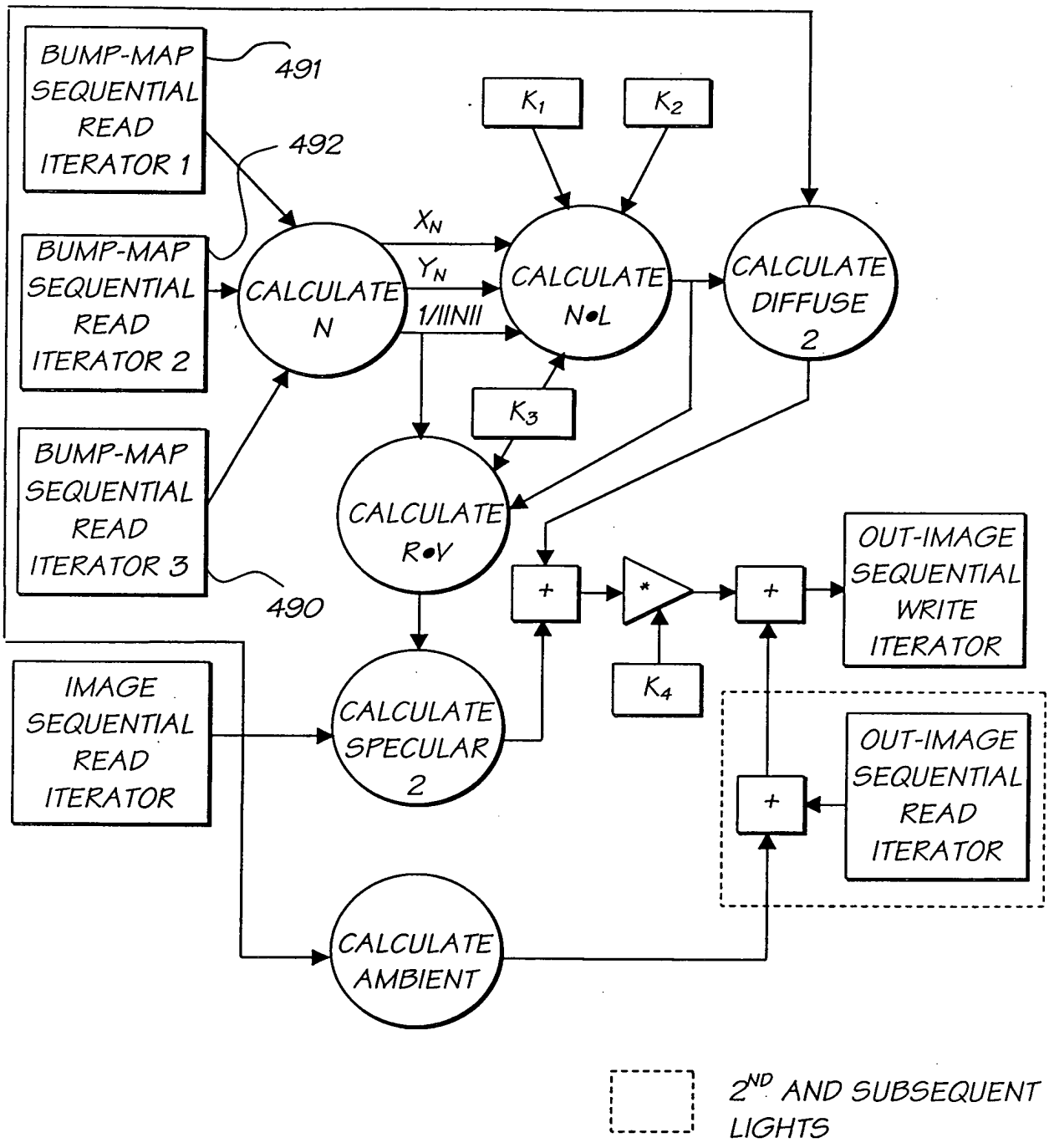

BUMP-MAP SEQUENTIAL READ ITERATOR 1
491
492
BUMP-MAP SEQUENTIAL READ ITERATOR 2
BUMP-MAP SEQUENTIAL READ ITERATOR 3
490
CALCULATE N
X_N
Y_N
1/||N||
K_1
K_2
CALCULATE N•L
CALCULATE DIFFUSE 2
K_3
CALCULATE R•V
+
*
+
OUT-IMAGE SEQUENTIAL WRITE ITERATOR
K_4
IMAGE SEQUENTIAL READ ITERATOR
CALCULATE SPECULAR 2
+
OUT-IMAGE SEQUENTIAL READ ITERATOR
CALCULATE AMBIENT
2ND AND SUBSEQUENT LIGHTS

FIG. 148

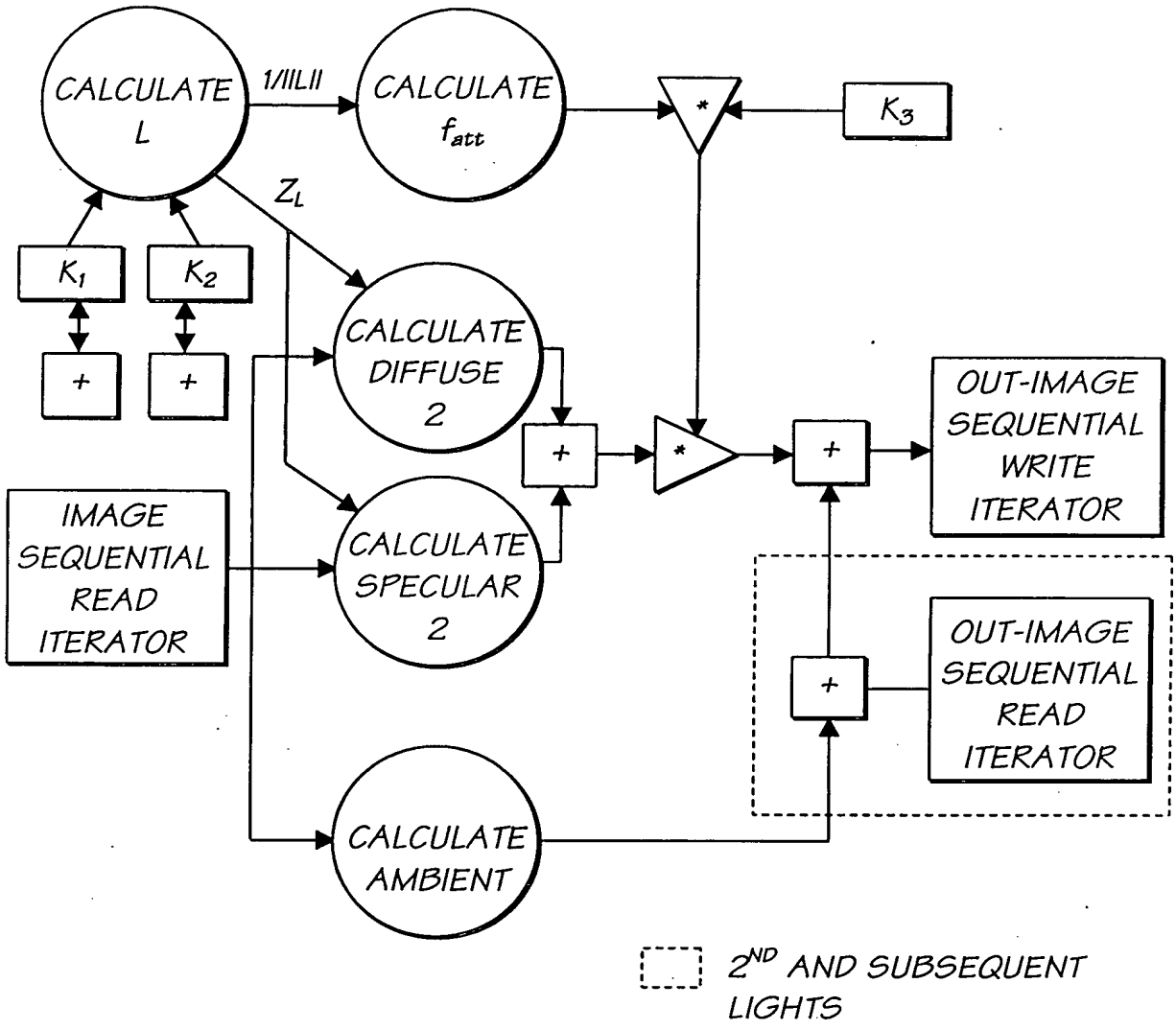

CALCULATE L
1/||L||
CALCULATE f_{att}
*
K_3
Z_L
K_1
K_2
+
+
CALCULATE DIFFUSE 2
+
*
+
OUT-IMAGE SEQUENTIAL WRITE ITERATOR
IMAGE SEQUENTIAL READ ITERATOR
CALCULATE SPECULAR 2
+
OUT-IMAGE SEQUENTIAL READ ITERATOR
CALCULATE AMBIENT
2ND AND SUBSEQUENT LIGHTS

FIG. 149

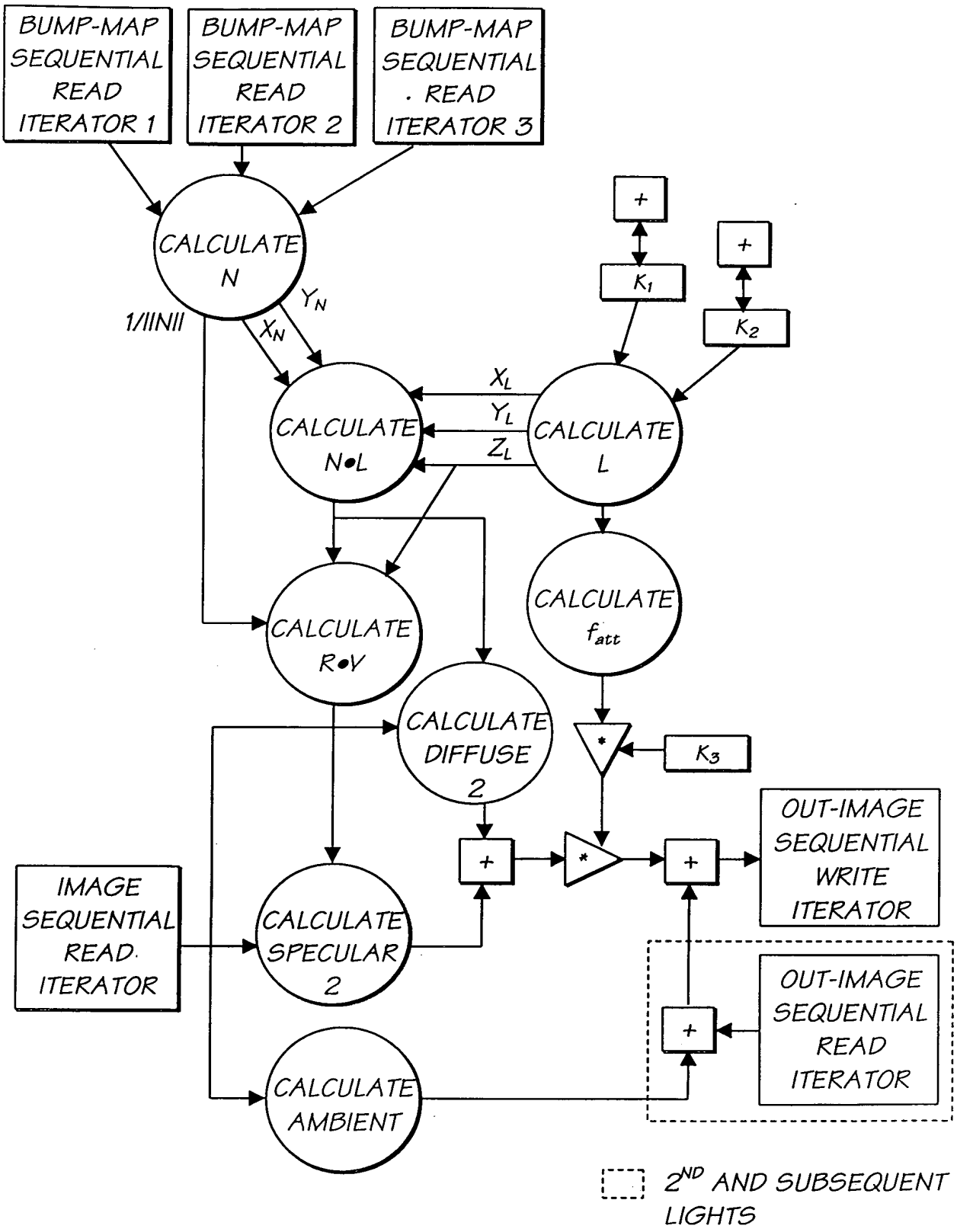

BUMP-MAP SEQUENTIAL READ ITERATOR 1
BUMP-MAP SEQUENTIAL READ ITERATOR 2
BUMP-MAP SEQUENTIAL . READ ITERATOR 3
CALCULATE N
1/||N||
Y_N
X_N
+
K_1
+
K_2
X_L
Y_L
Z_L
CALCULATE N•L
CALCULATE L
CALCULATE R•V
CALCULATE f_{att}
CALCULATE DIFFUSE 2
*
K_3
+
*
+
OUT-IMAGE SEQUENTIAL WRITE ITERATOR
IMAGE SEQUENTIAL READ ITERATOR
CALCULATE SPECULAR 2
+
OUT-IMAGE SEQUENTIAL READ ITERATOR
CALCULATE AMBIENT
2^{ND} AND SUBSEQUENT LIGHTS

FIG. 150

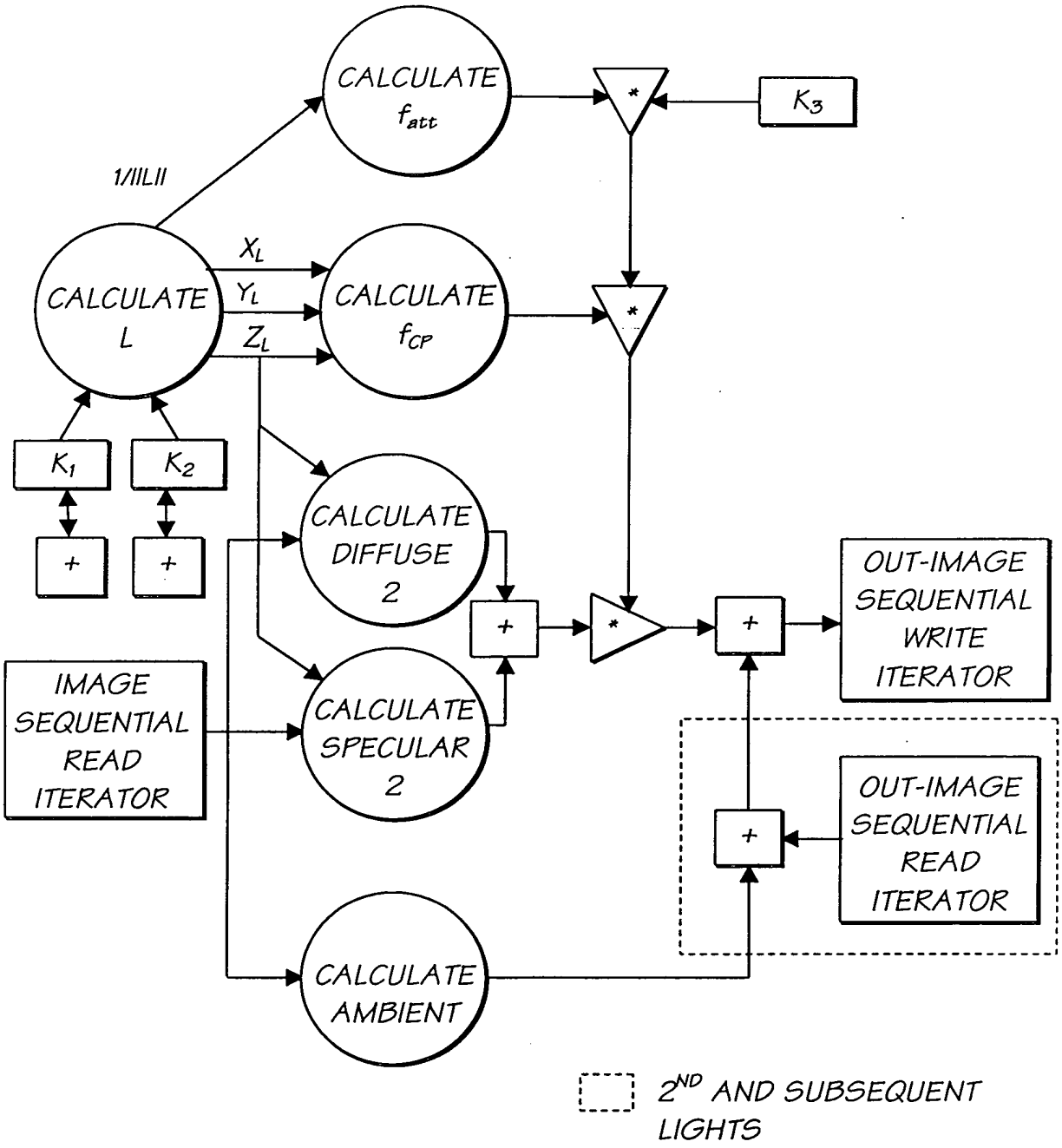

CALCULATE f_{att}
K_3
1/IIILII
CALCULATE L
X_L
Y_L
Z_L
CALCULATE f_{CP}
K_1
K_2
+
+
CALCULATE DIFFUSE 2
+
*
+
OUT-IMAGE SEQUENTIAL WRITE ITERATOR
IMAGE SEQUENTIAL READ ITERATOR
CALCULATE SPECULAR 2
OUT-IMAGE SEQUENTIAL READ ITERATOR
+
CALCULATE AMBIENT
2ND AND SUBSEQUENT LIGHTS

FIG. 151

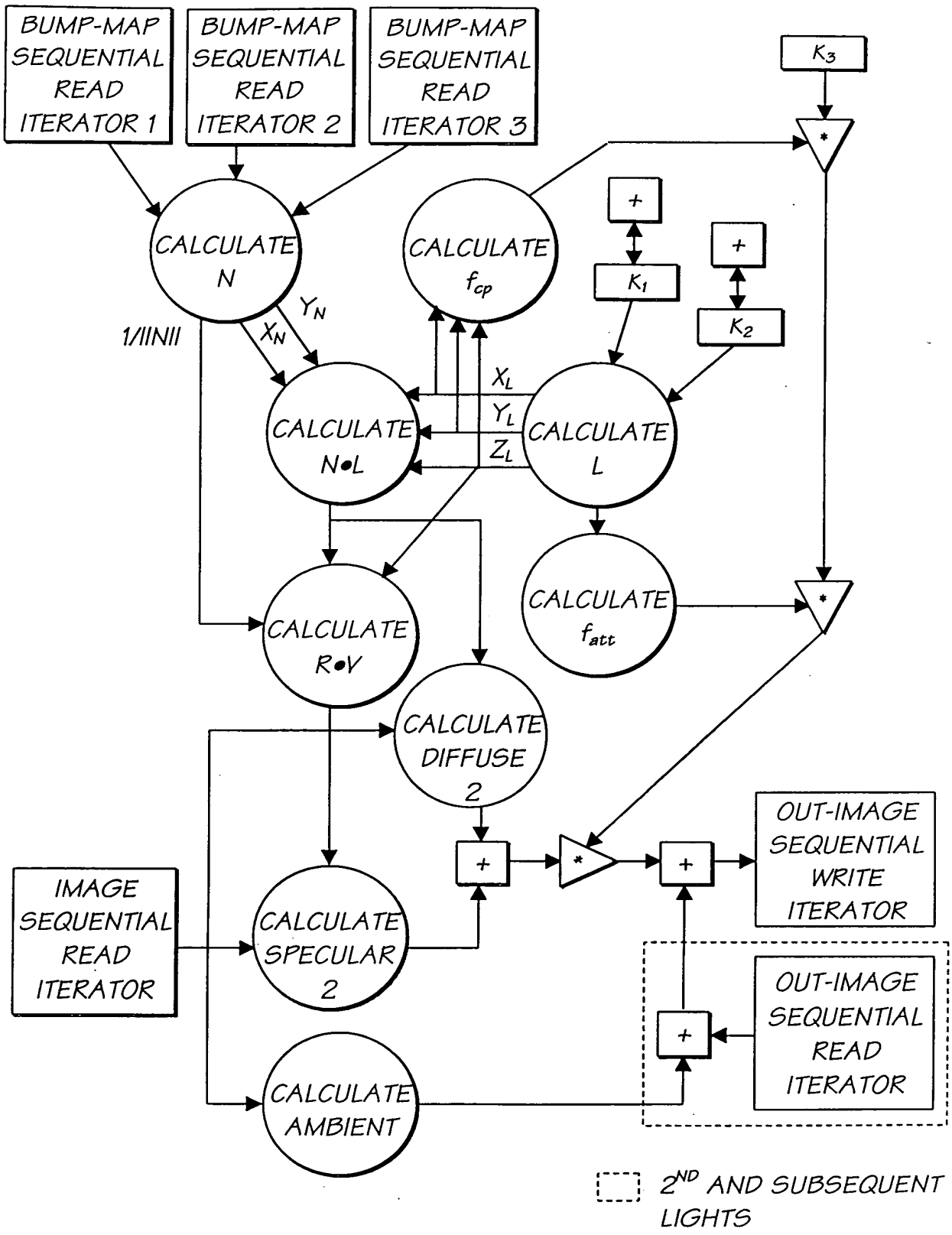

BUMP-MAP SEQUENTIAL READ ITERATOR 1
BUMP-MAP SEQUENTIAL READ ITERATOR 2
BUMP-MAP SEQUENTIAL READ ITERATOR 3
K3
CALCULATE N
CALCULATE fcp
K1
K2
YN
XN
1/||N||
XL
YL
ZL
CALCULATE N•L
CALCULATE L
CALCULATE R•V
CALCULATE fatt
CALCULATE DIFFUSE 2
OUT-IMAGE SEQUENTIAL WRITE ITERATOR
IMAGE SEQUENTIAL READ ITERATOR
CALCULATE SPECULAR 2
OUT-IMAGE SEQUENTIAL READ ITERATOR
CALCULATE AMBIENT
2ND AND SUBSEQUENT LIGHTS

FIG. 152

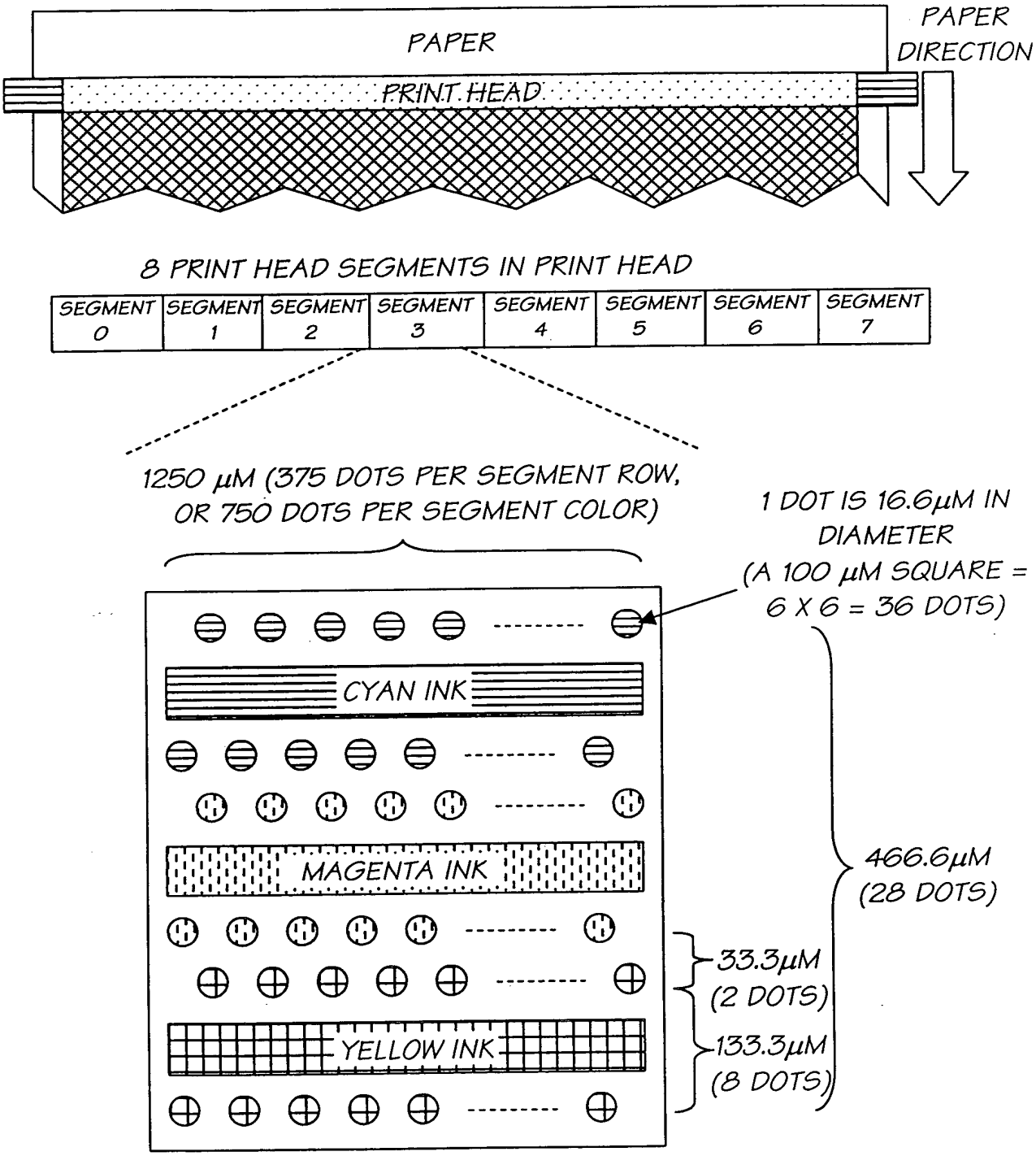

PAPER
PAPER DIRECTION
PRINT HEAD
8 PRINT HEAD SEGMENTS IN PRINT HEAD
SEGMENT 0
SEGMENT 1
SEGMENT 2
SEGMENT 3
SEGMENT 4
SEGMENT 5
SEGMENT 6
SEGMENT 7
1250 μM (375 DOTS PER SEGMENT ROW, OR 750 DOTS PER SEGMENT COLOR)
1 DOT IS 16.6μM IN DIAMETER (A 100 μM SQUARE = 6 X 6 = 36 DOTS)
CYAN INK
MAGENTA INK
YELLOW INK
466.6μM (28 DOTS)
33.3μM (2 DOTS)
133.3μM (8 DOTS)
EACH SEGMENT CONTAINS 6 ROWS OF DOTS: ODD AND EVEN CYAN, MAGENTA, AND YELLOW.

FIG. 153

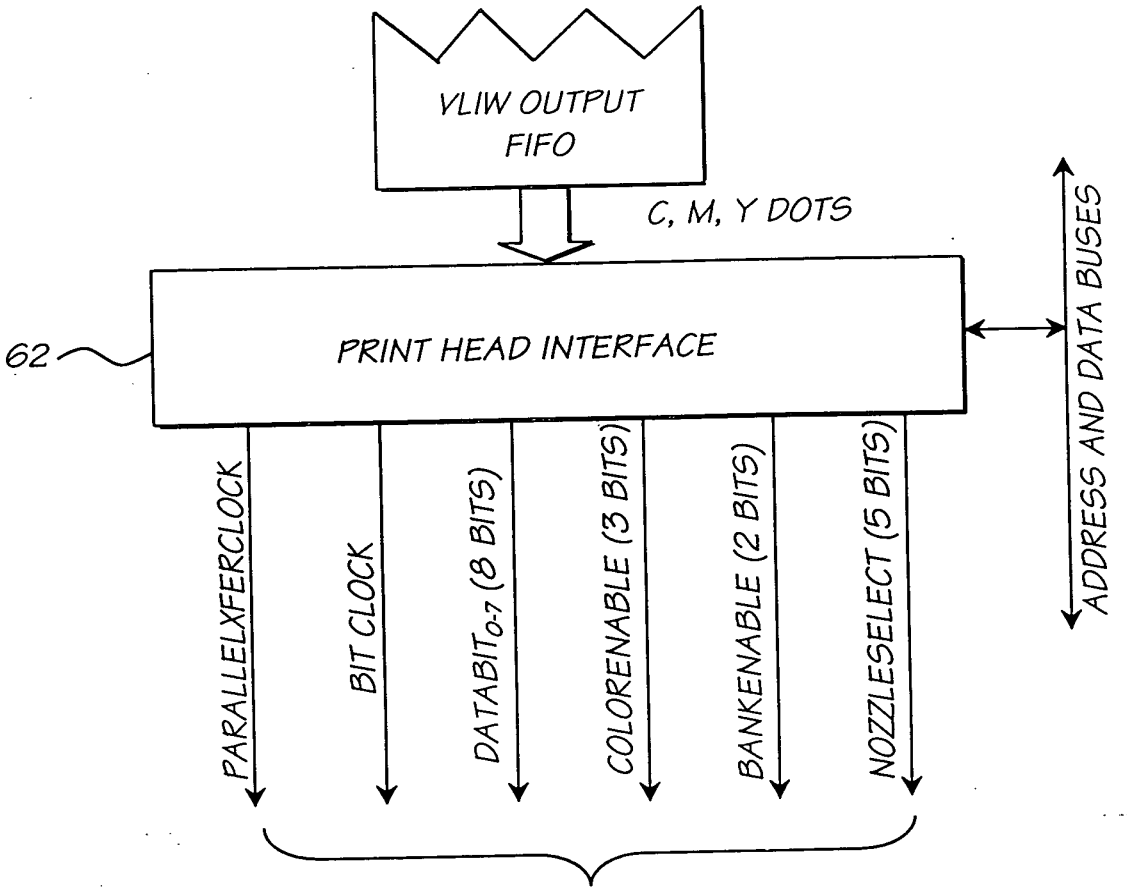


FIG. 154

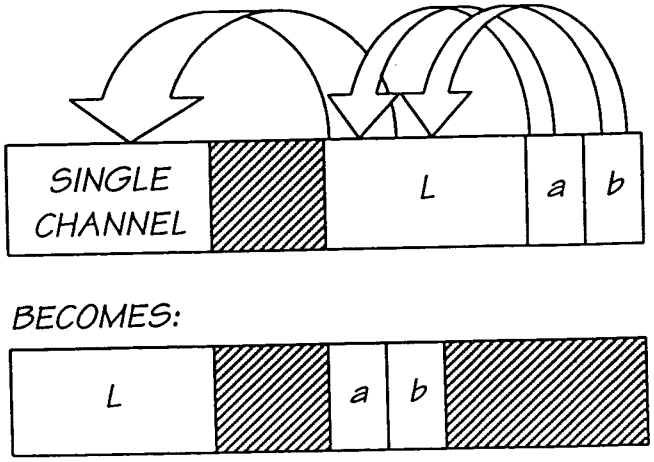


FIG. 155

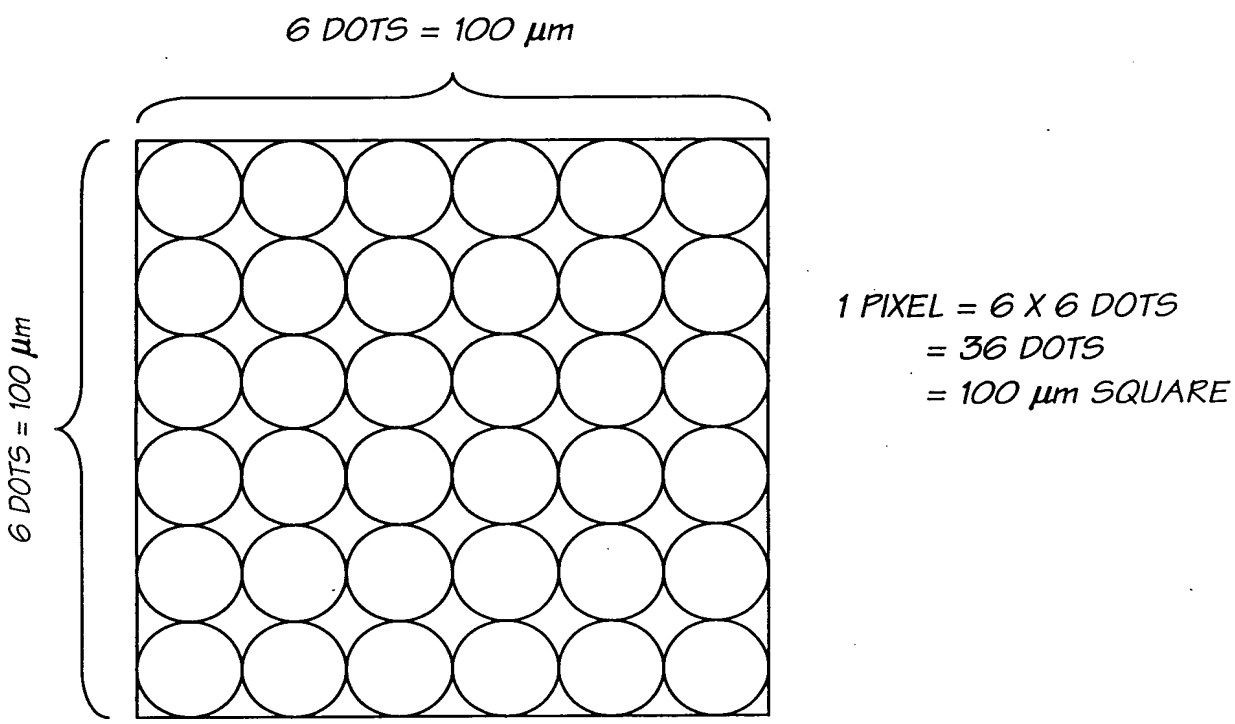

6 DOTS = 100 μm
6 DOTS = 100 μm
1 PIXEL = 6 X 6 DOTS
= 36 DOTS
= 100 μm SQUARE

FIG. 156

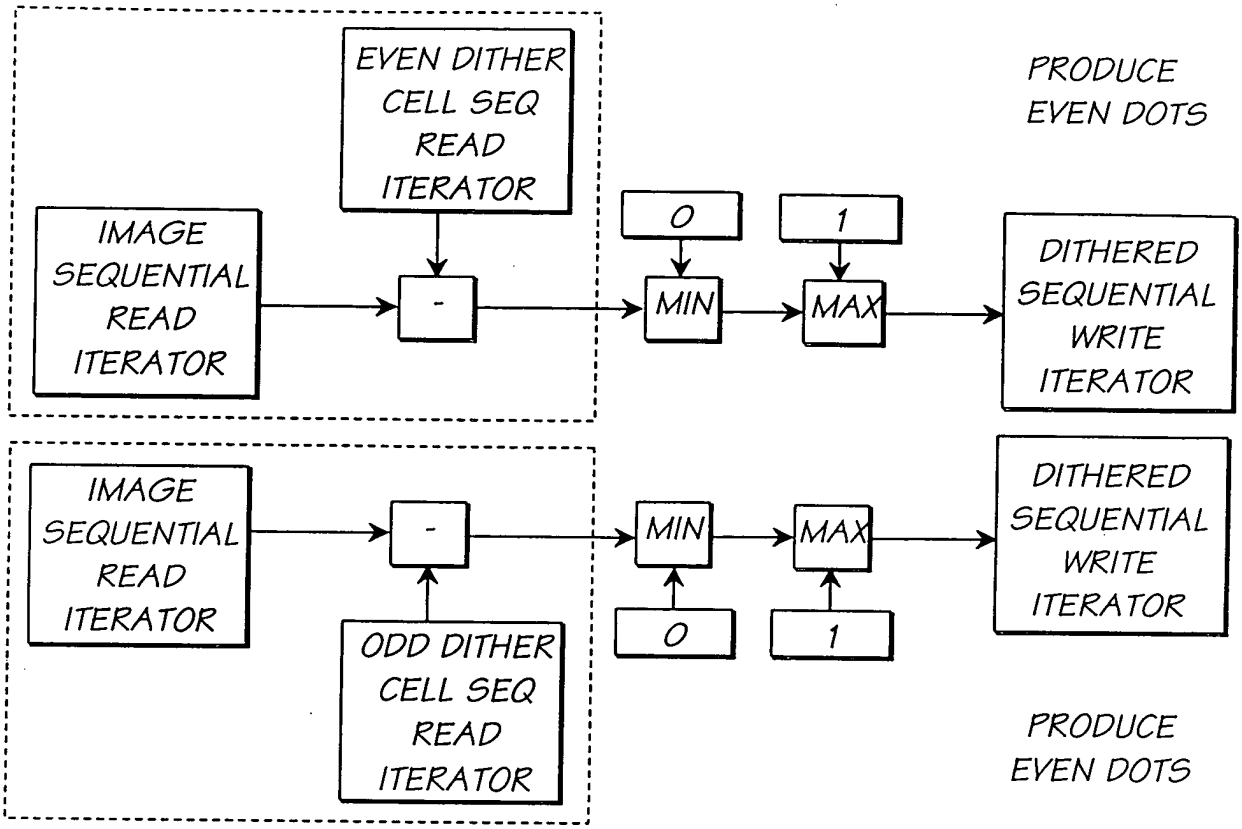


FIG. 157

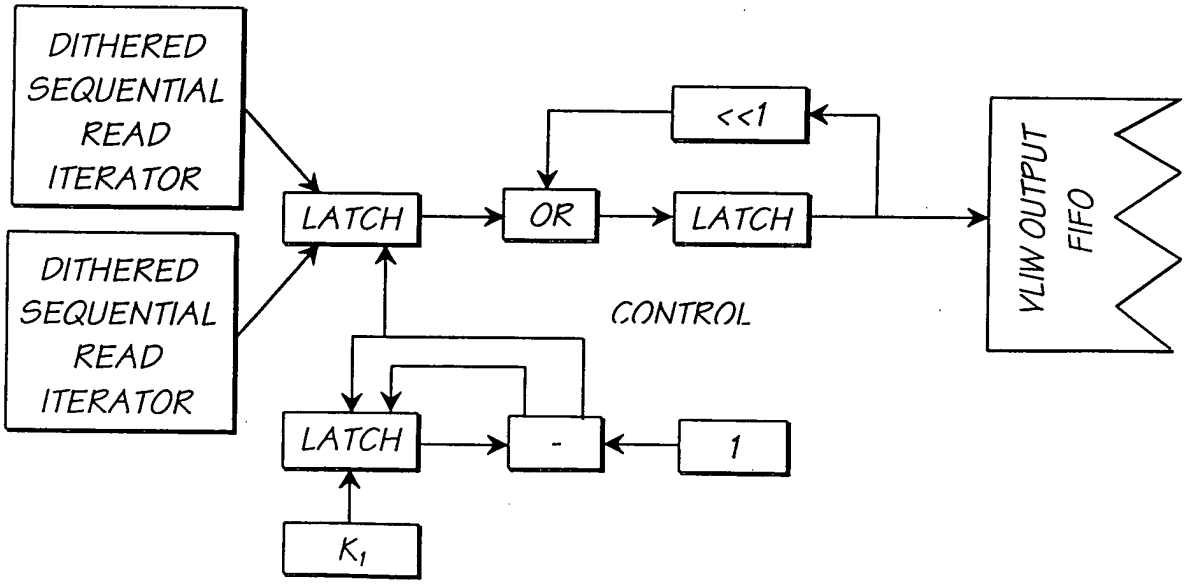


FIG. 158

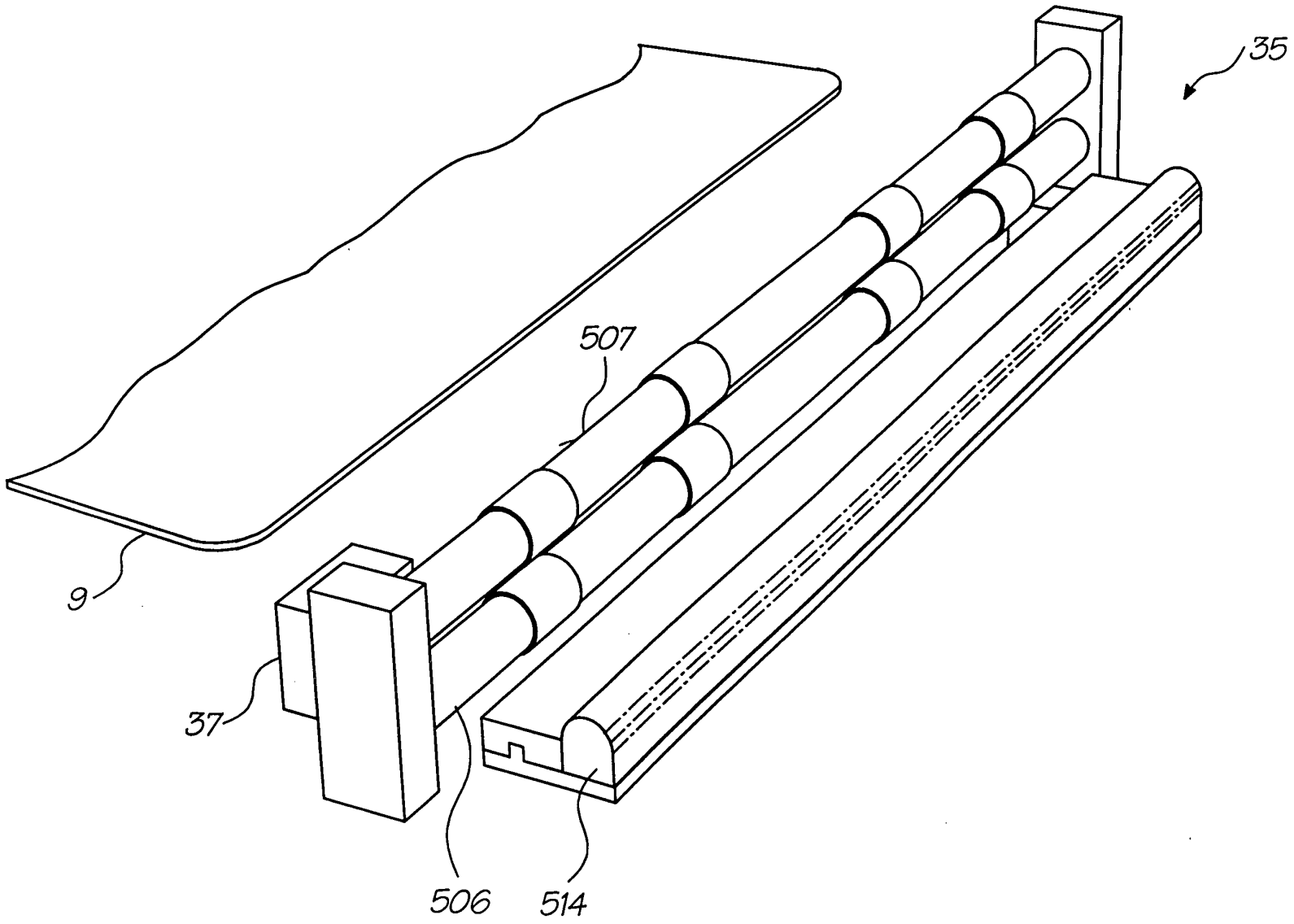


FIG. 159

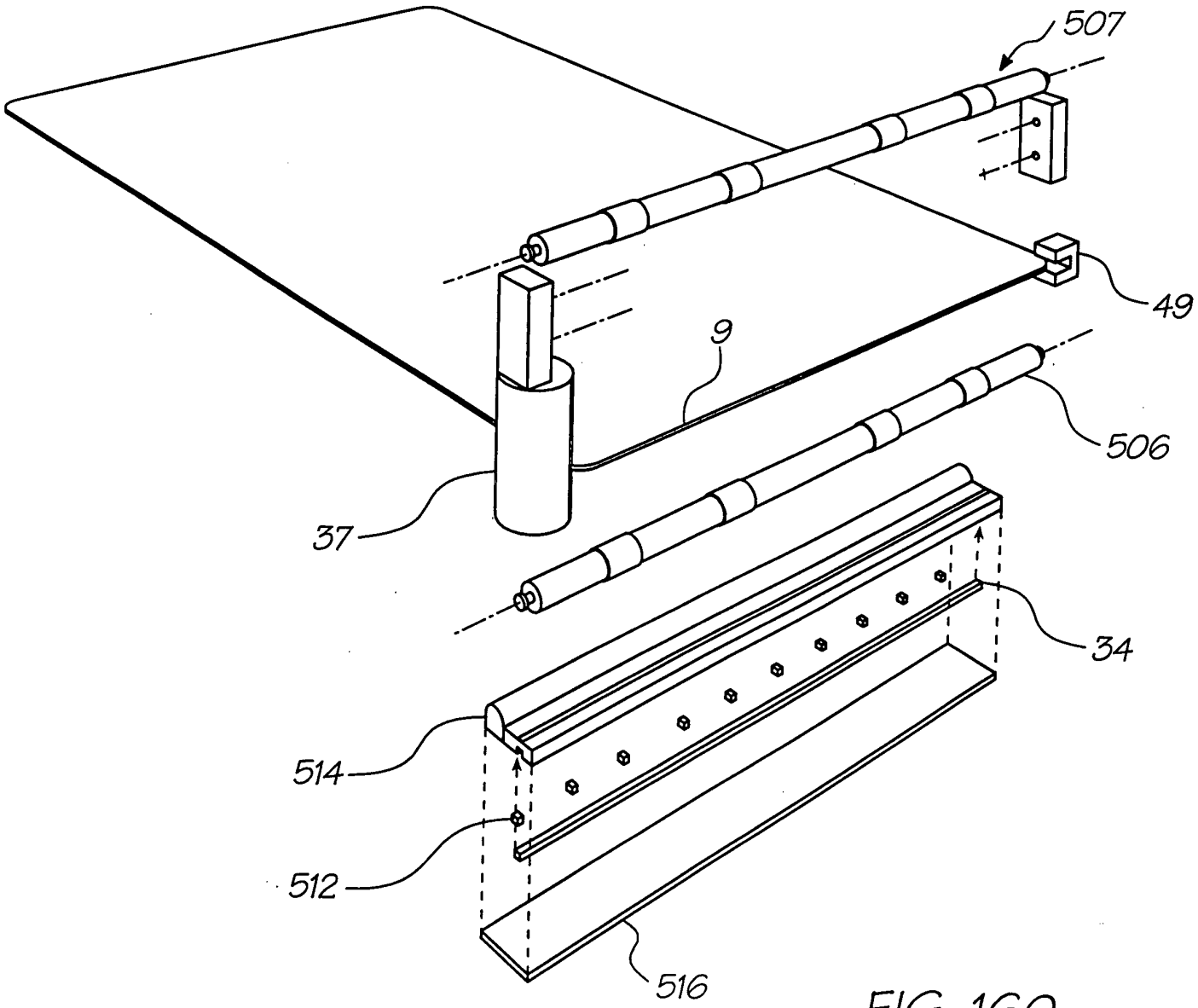


FIG. 160

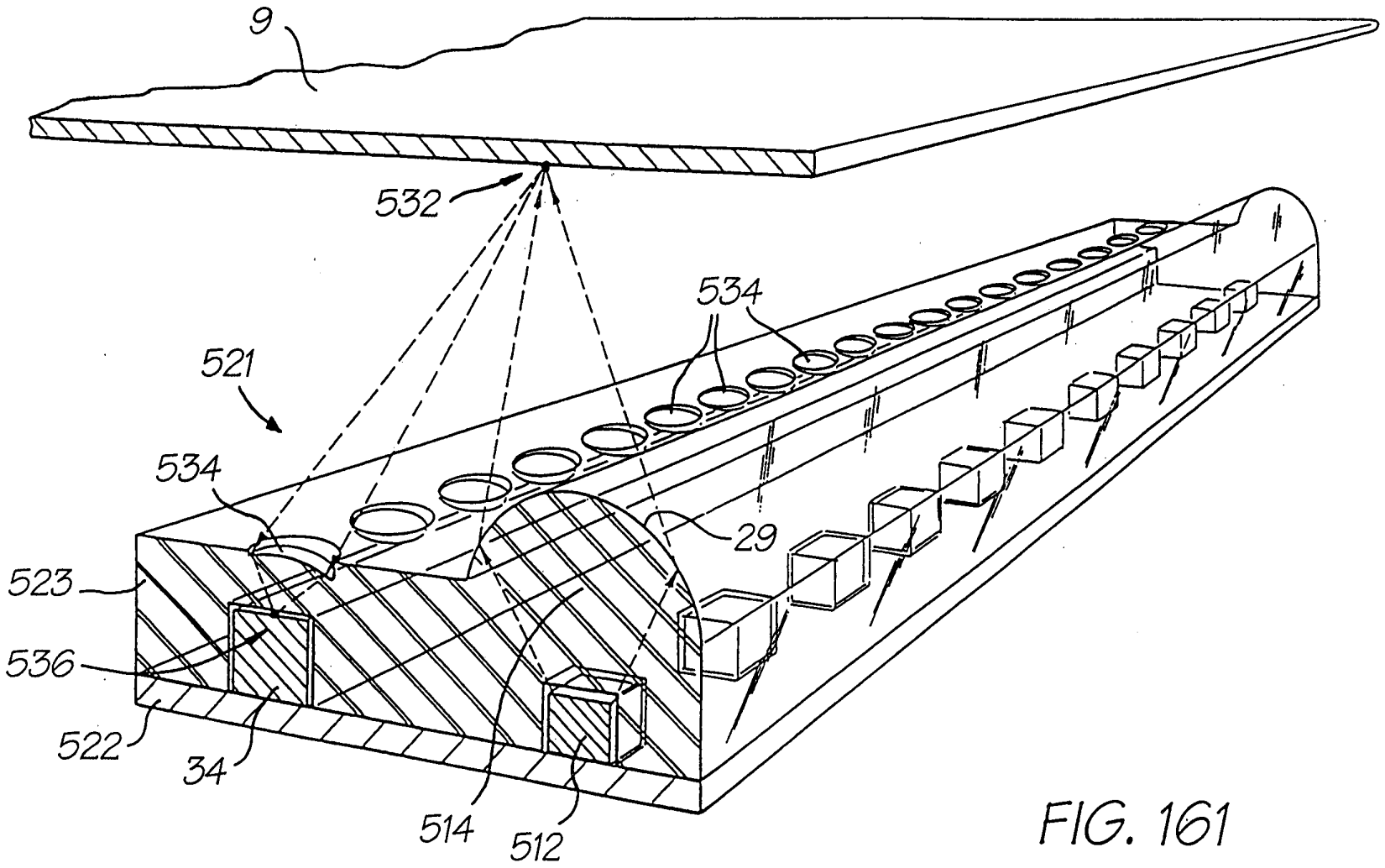


FIG. 161

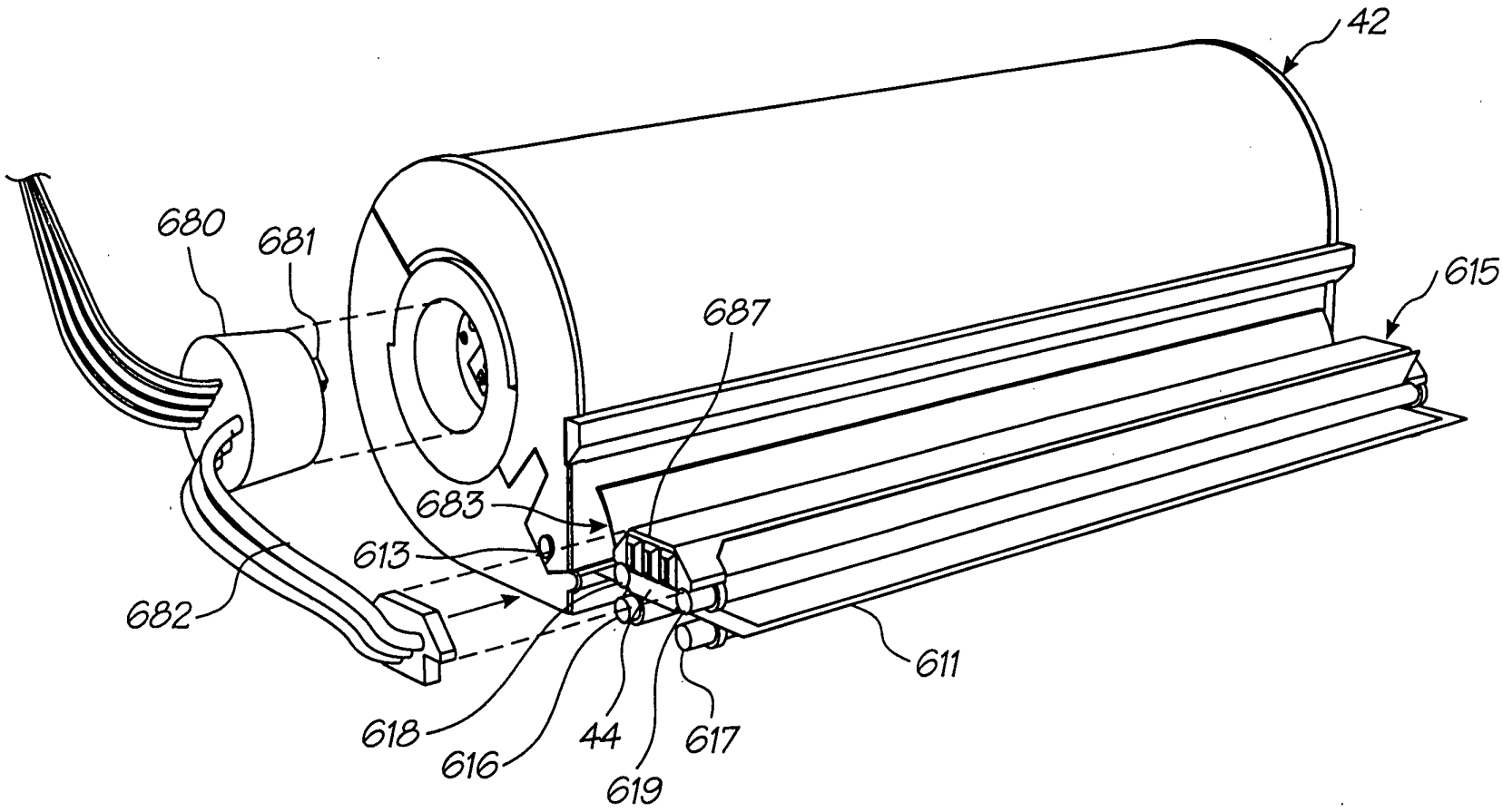


FIG. 162

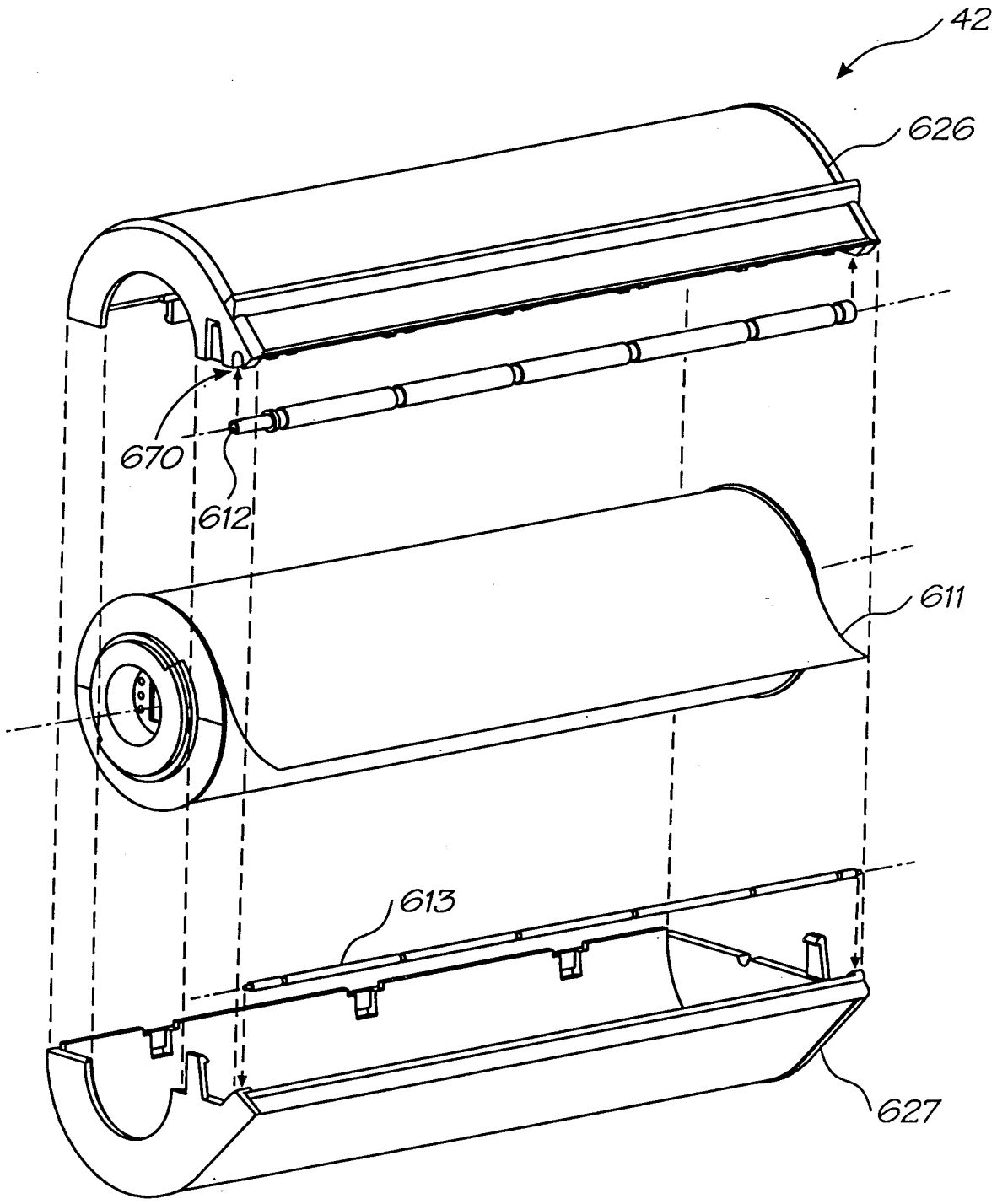


FIG. 163

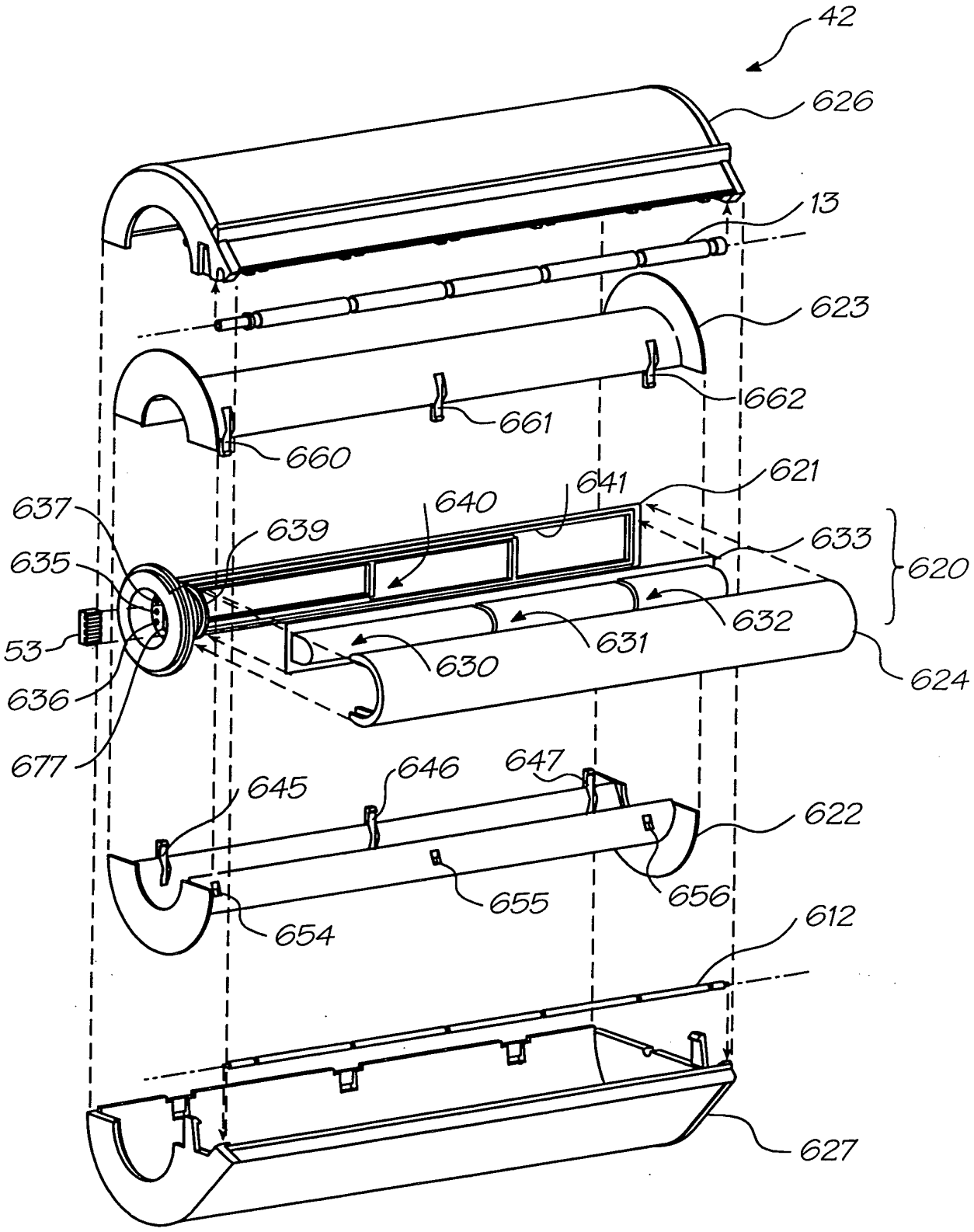

42
626
13
623
662
661
660
621
641
640
639
637
633
635
620
632
631
53
630
624
636
677
646
647
645
622
655
656
654
612
627

FIG. 164

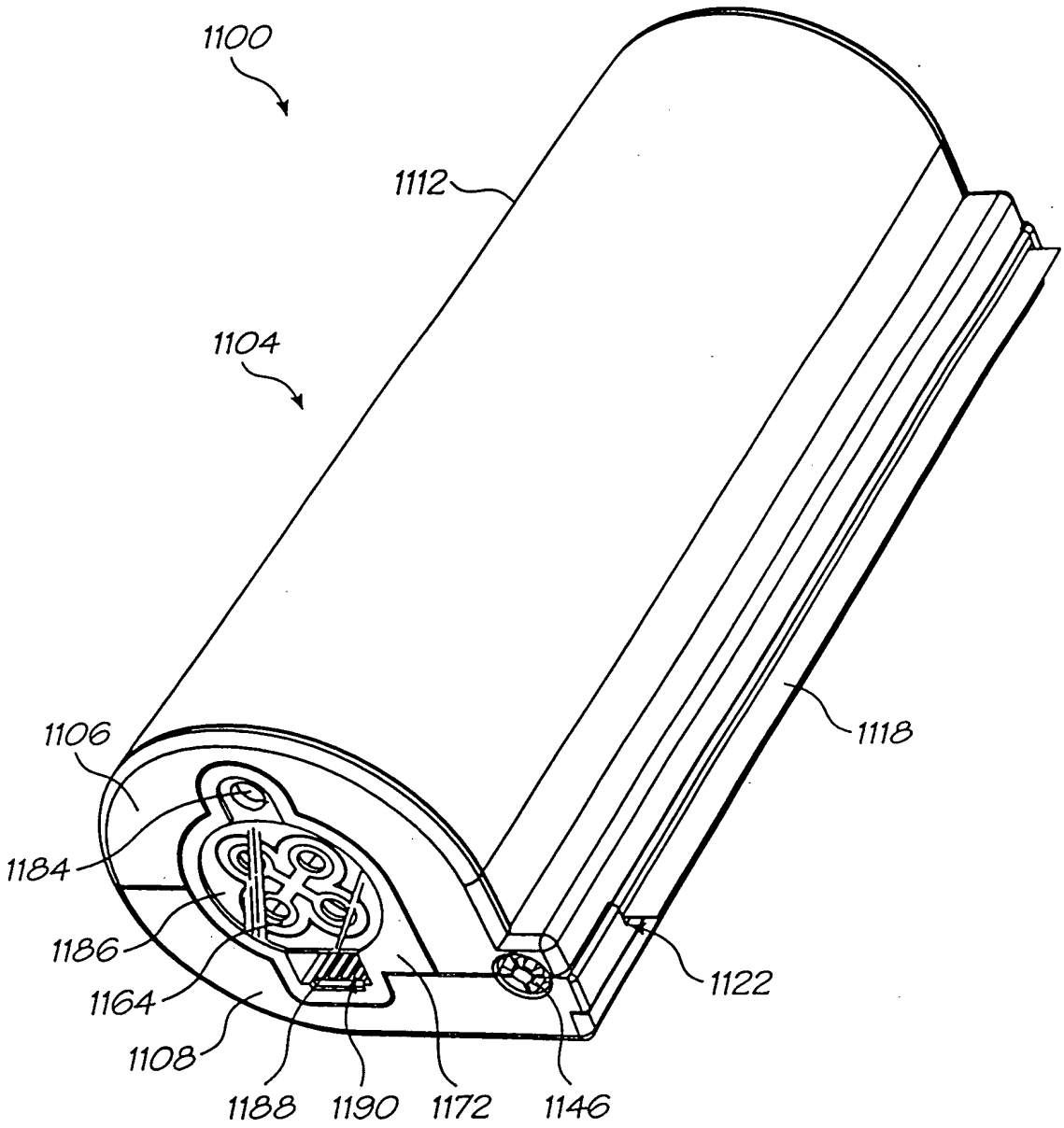

1100
1112
1104
1106
1118
1184
1186
1164
1122
1108
1188
1190
1172
1146

FIG. 164A

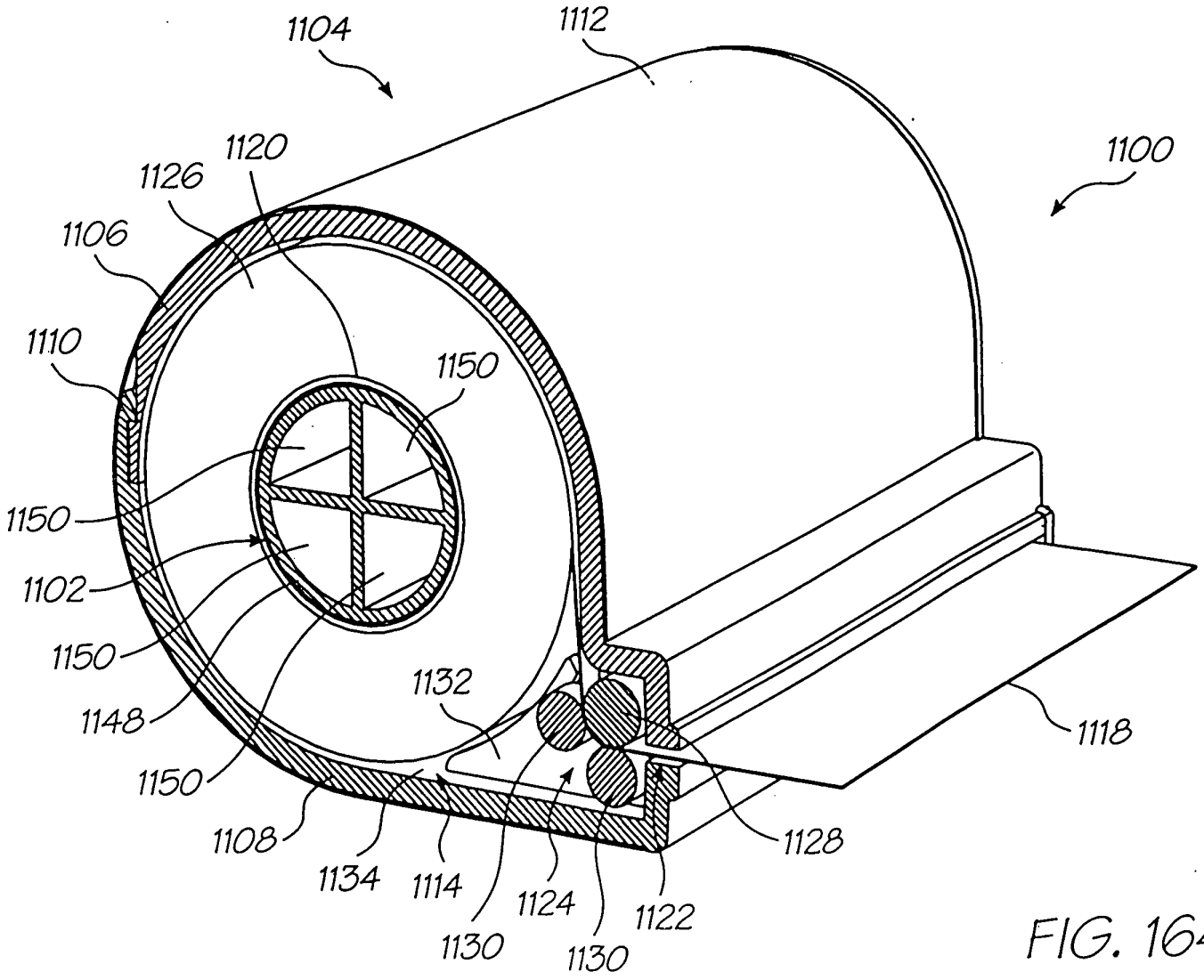

1104
1112
1120
1126
1100
1106
1110
1150
1150
1102
1150
1148
1132
1150
1118
1108
1128
1134
1114
1124
1122
1130
1130

FIG. 164B

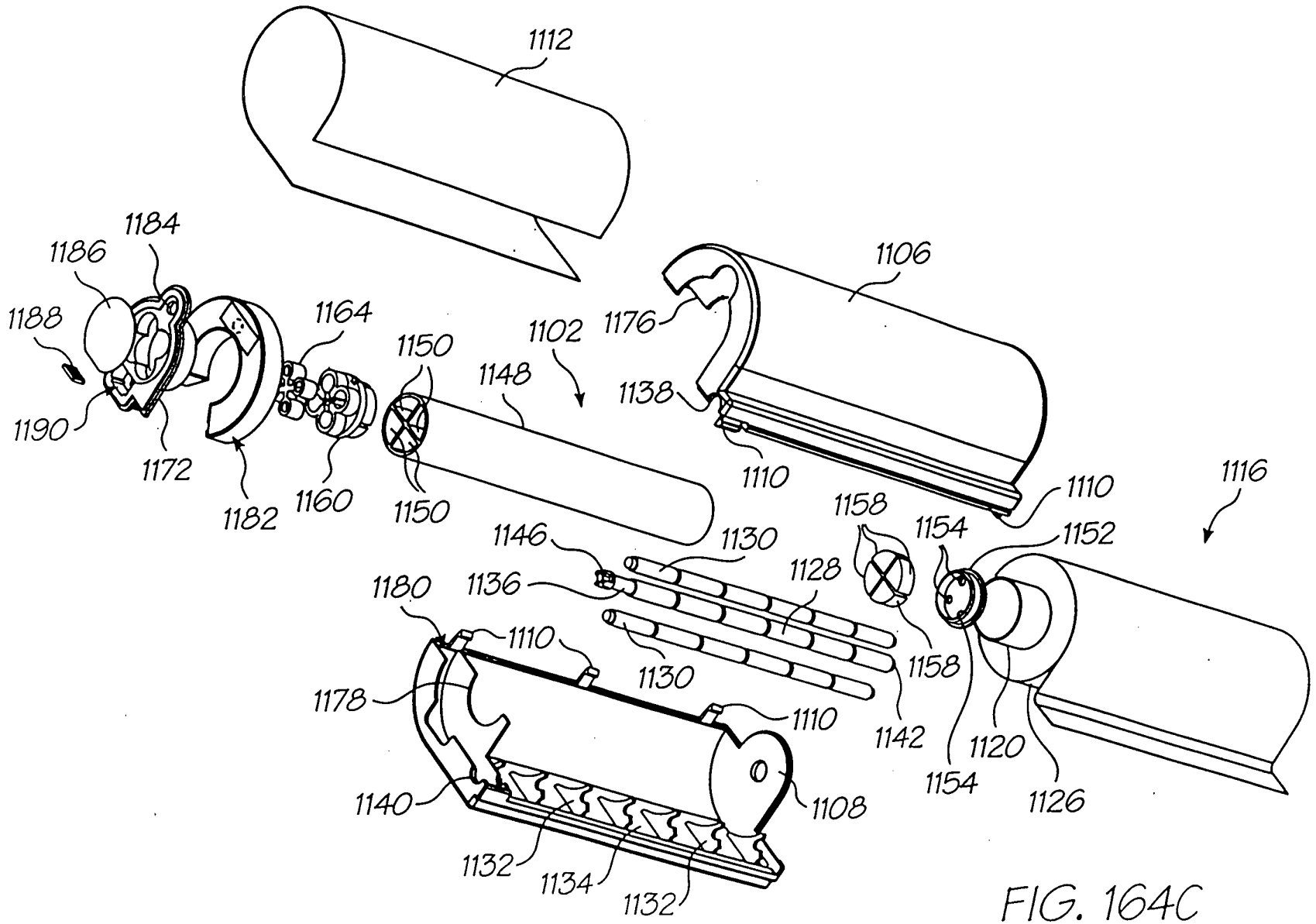

1112
1184
1186
1188
1190
1172
1182
1160
1164
1150
1148
1102
1176
1106
1138
1110
1158
1154
1152
1116
1146
1130
1128
1136
1180
1178
1140
1132
1134
1108
1120
1126
1142

FIG. 164C

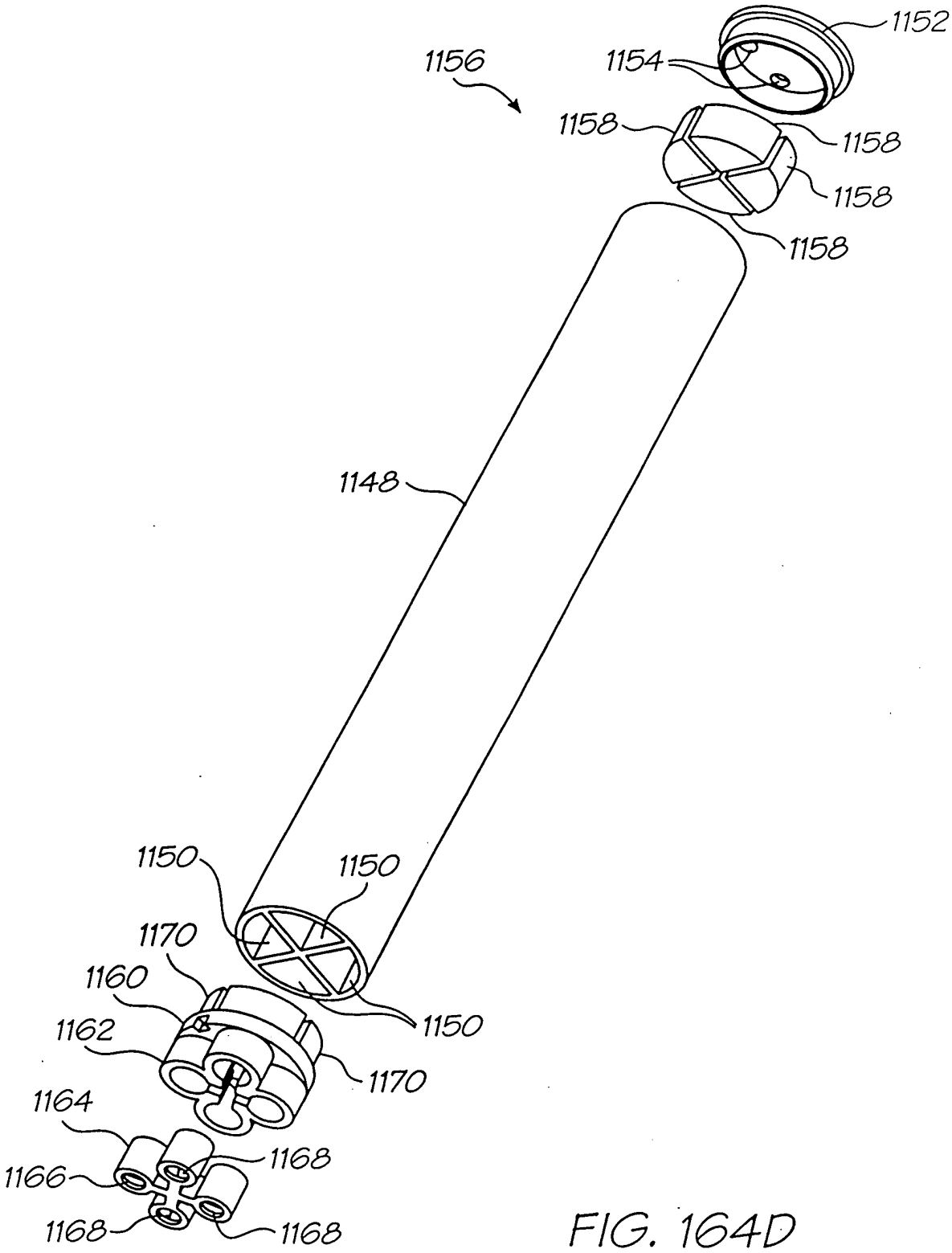

1152
1156
1154
1158
1158
1158
1158
1148
1150
1150
1170
1160
1150
1162
1170
1164
1168
1166
1168
1168

FIG. 164D

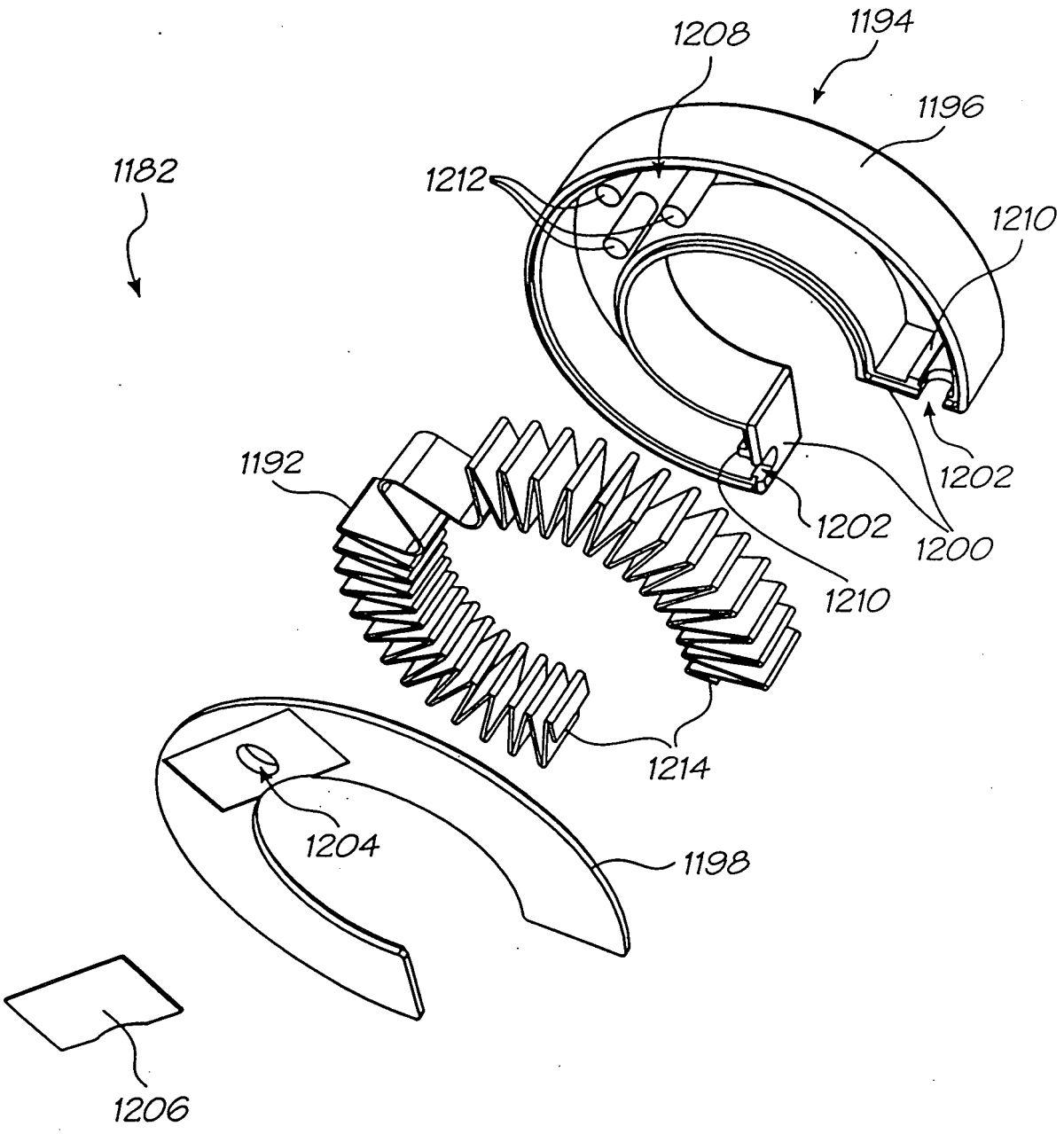

1208
1194
1196
1182
1212
1210
1192
1202
1202
1200
1210
1214
1204
1198
1206

FIG. 164E

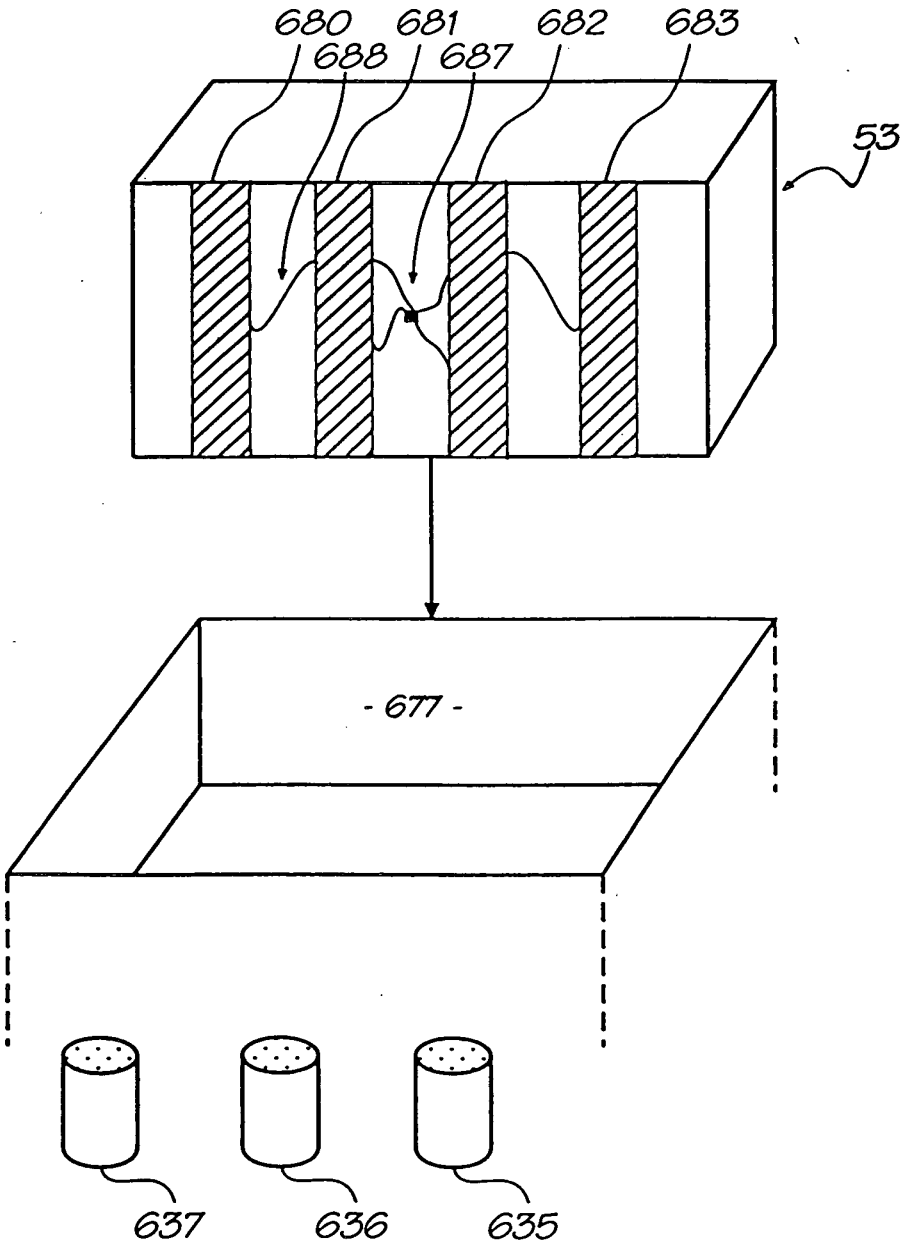


FIG. 165

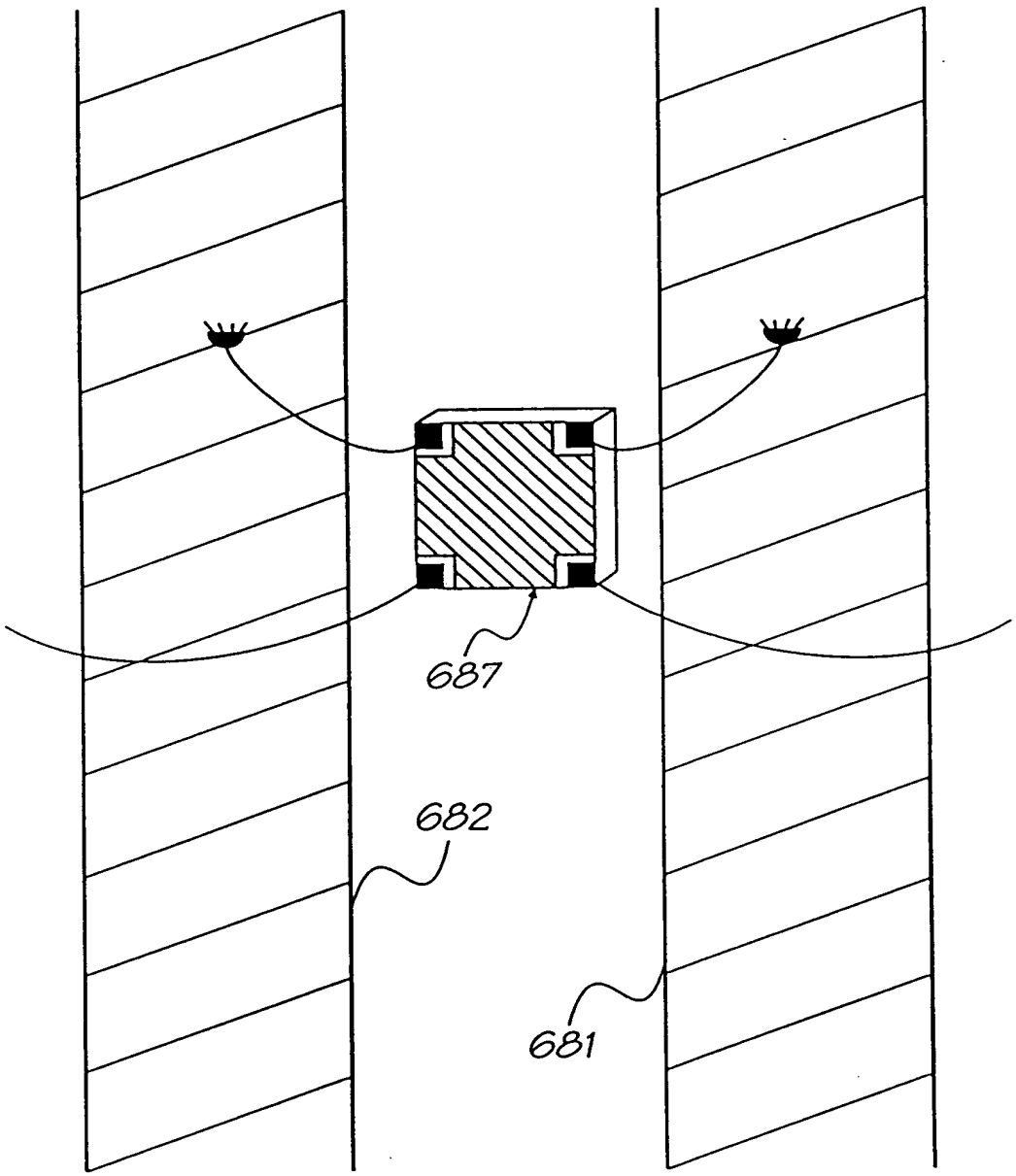


FIG. 166

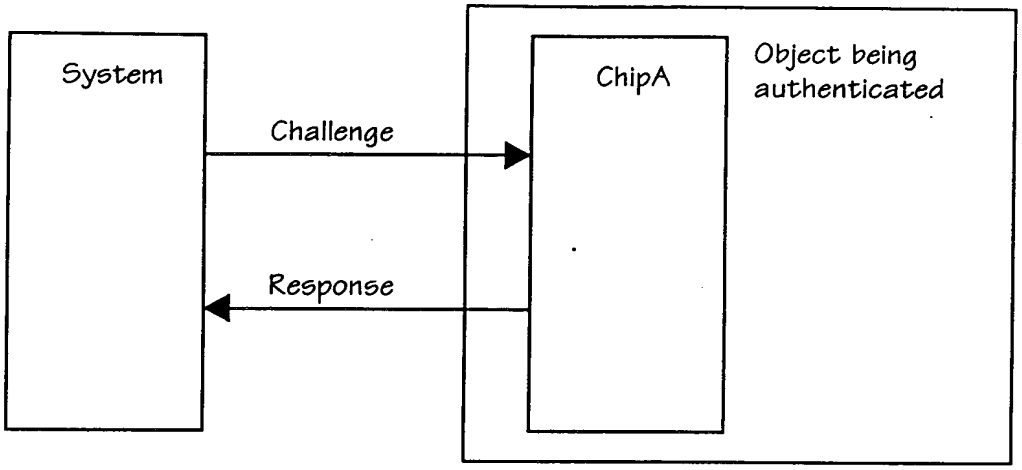


FIG. 167

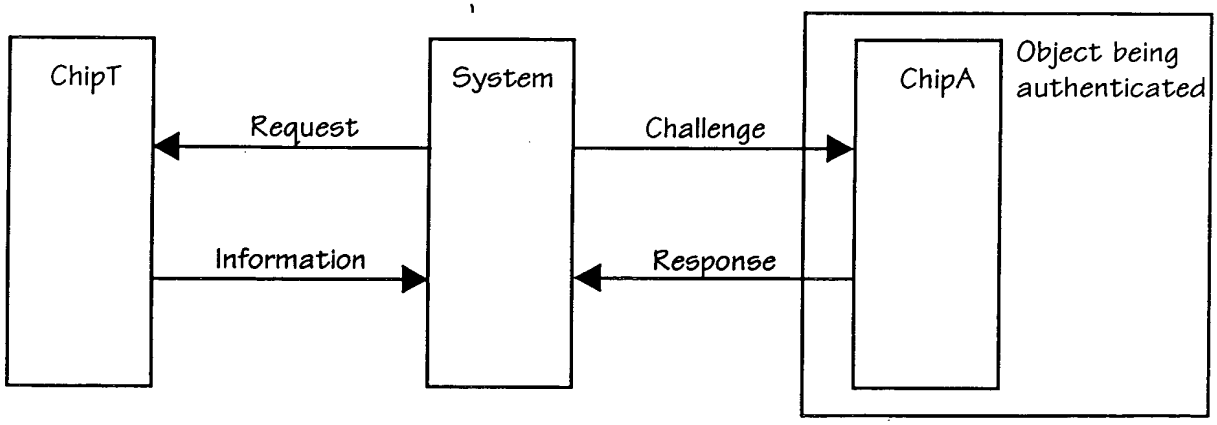


FIG. 168

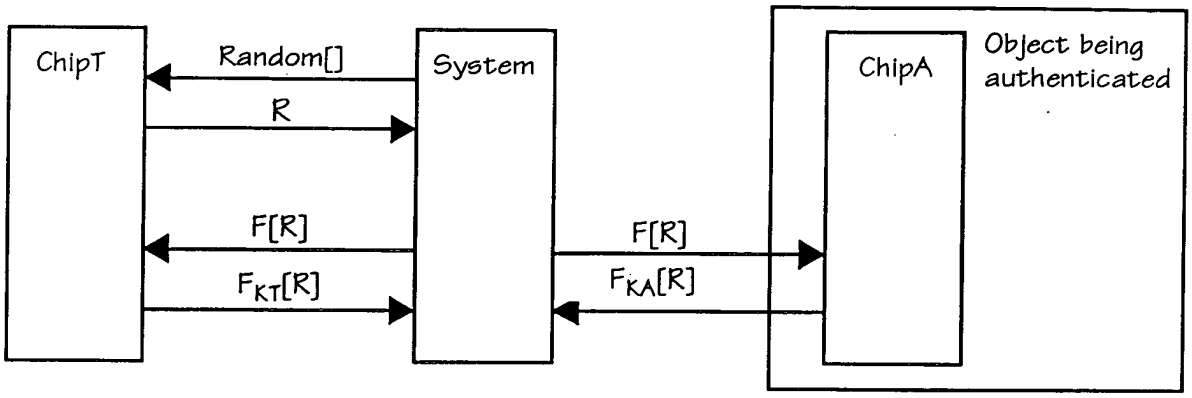


FIG. 169

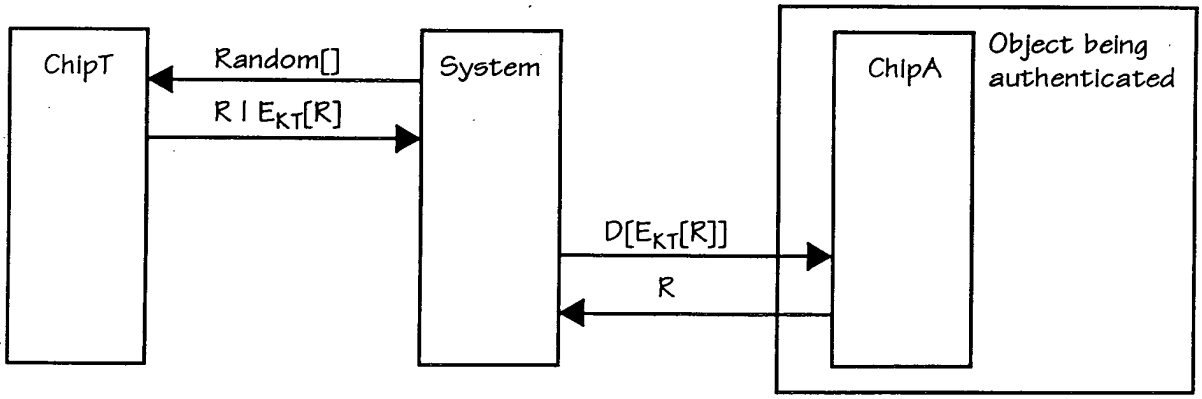


FIG. 170

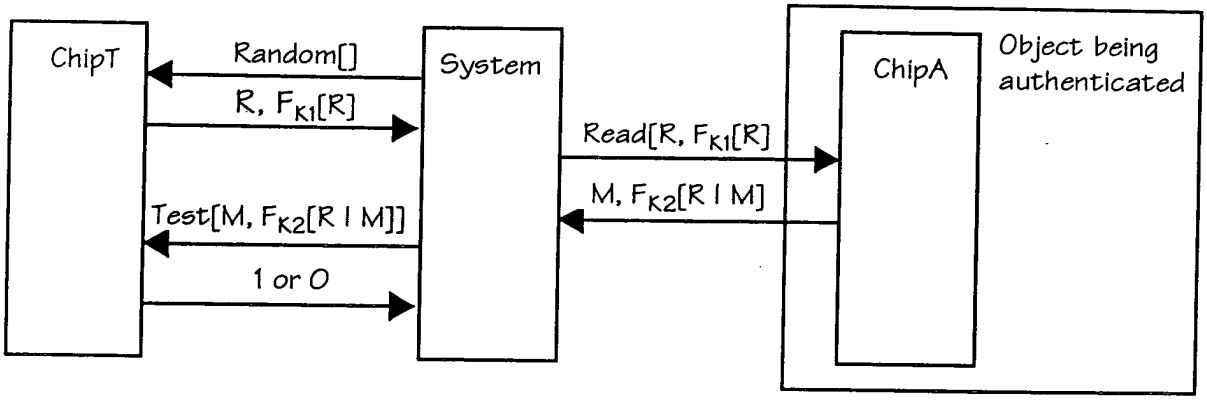


FIG. 171

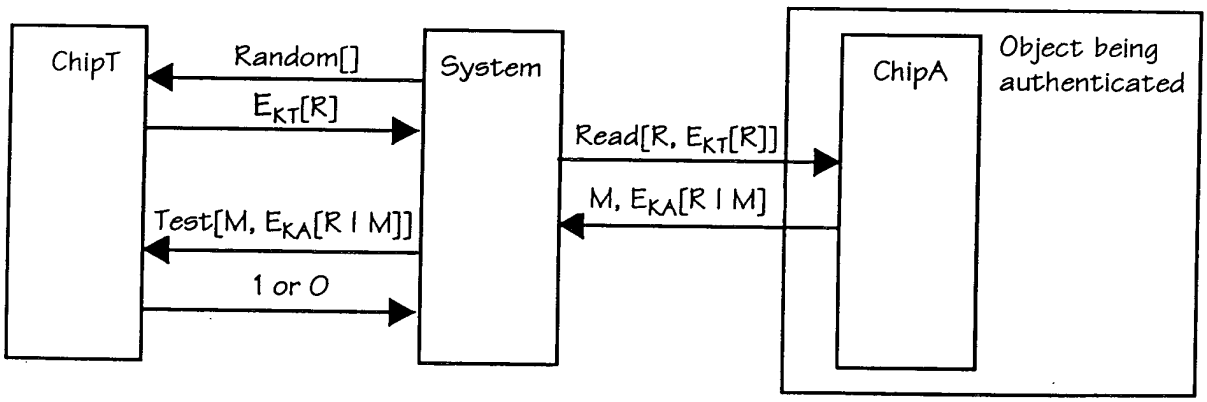


FIG. 172

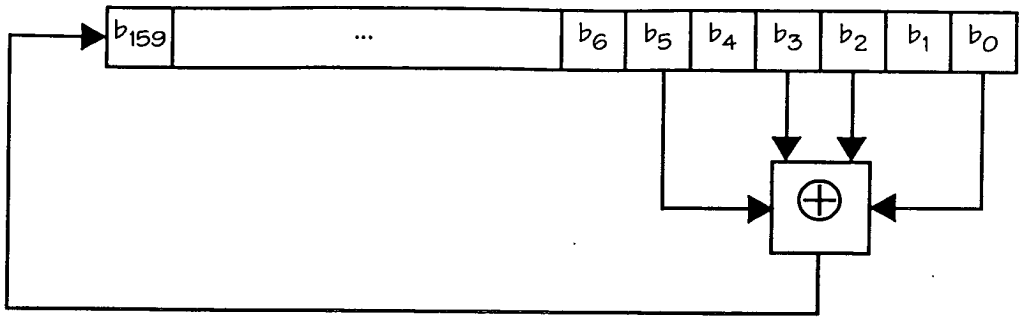


FIG. 173

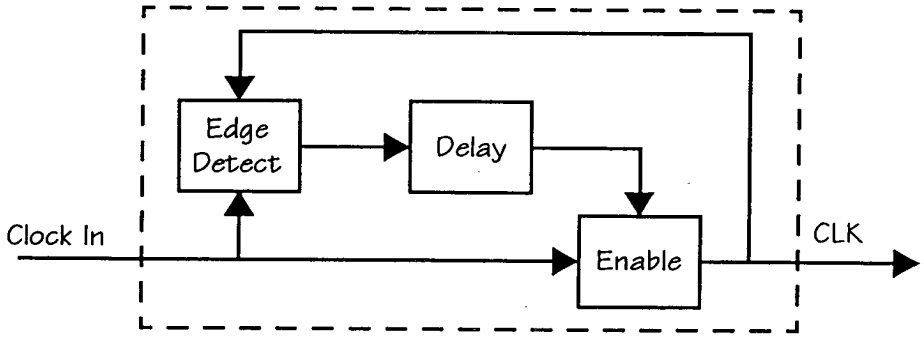


FIG. 174

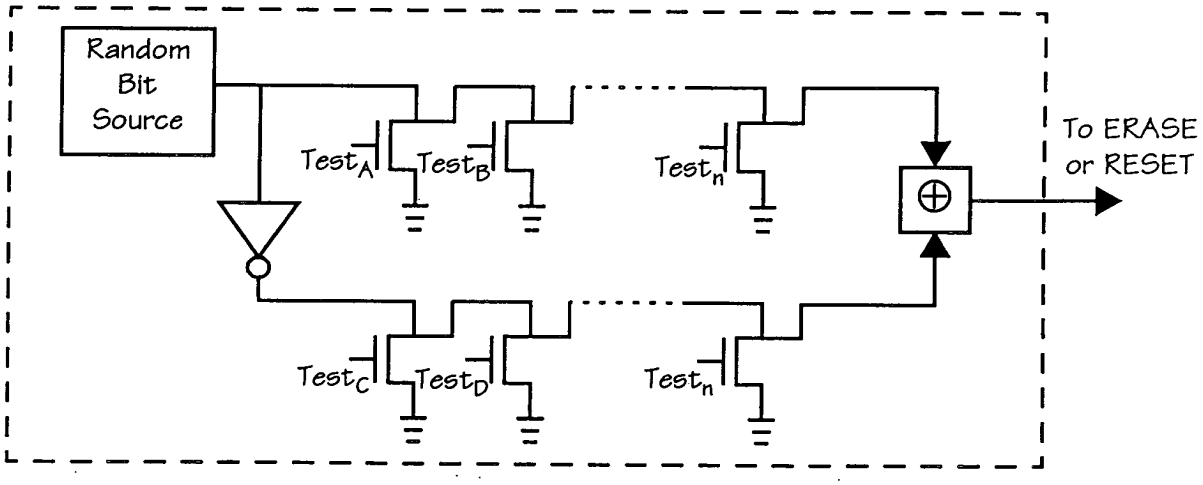


FIG. 175

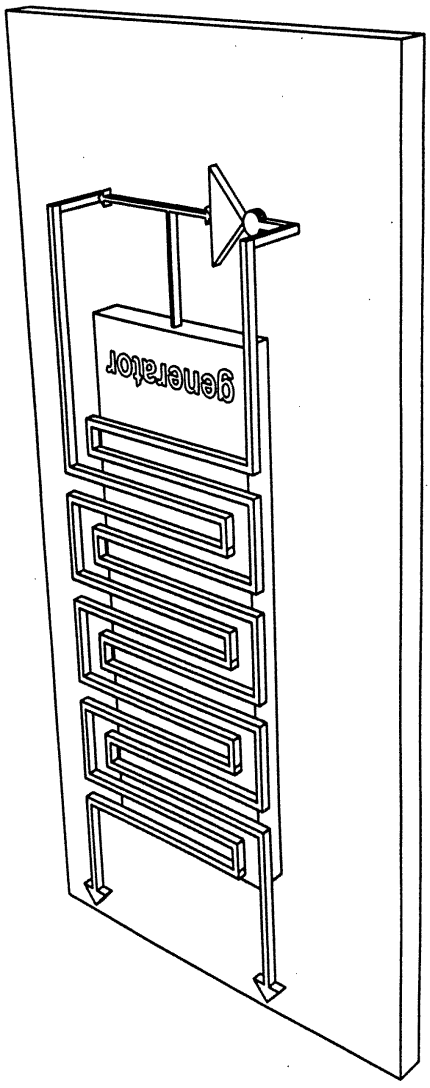


FIG. 17B

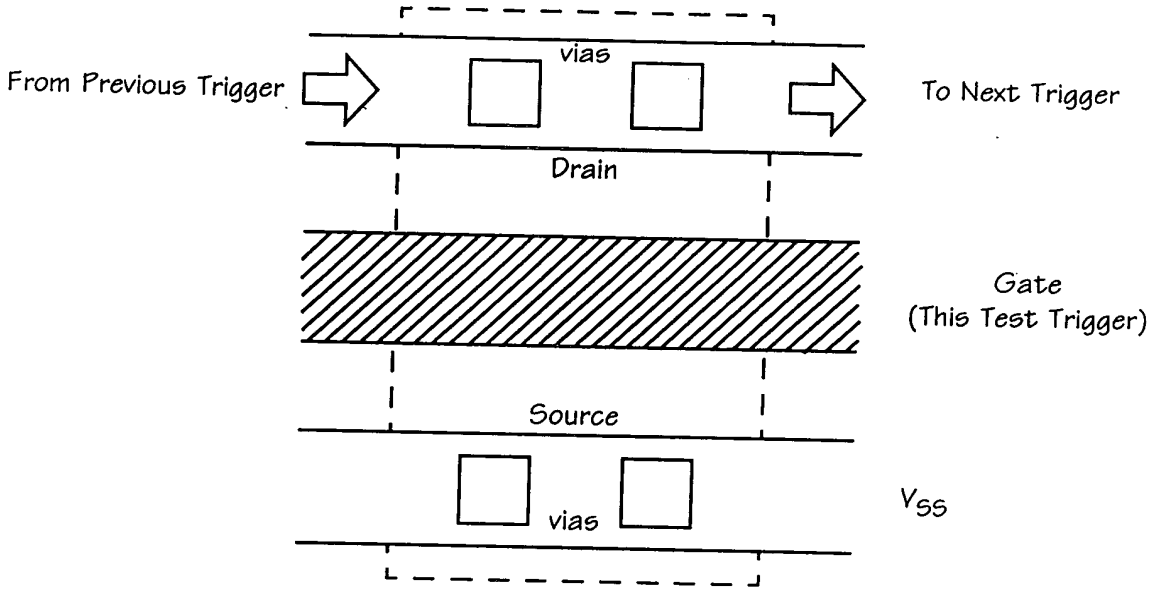


FIG. 176

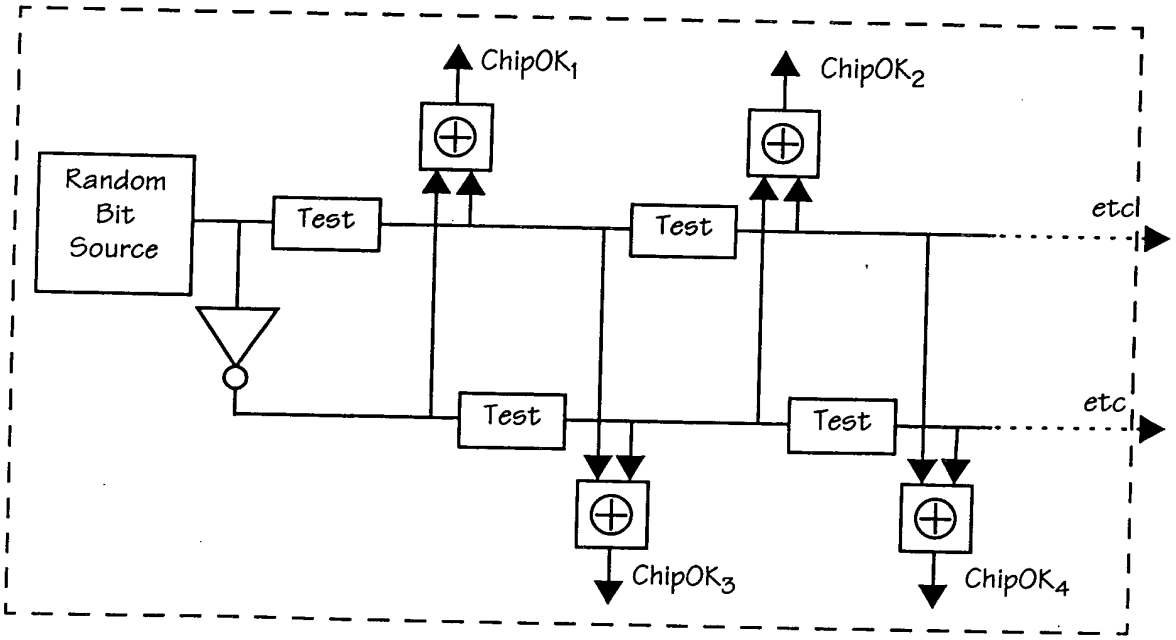


FIG. 177

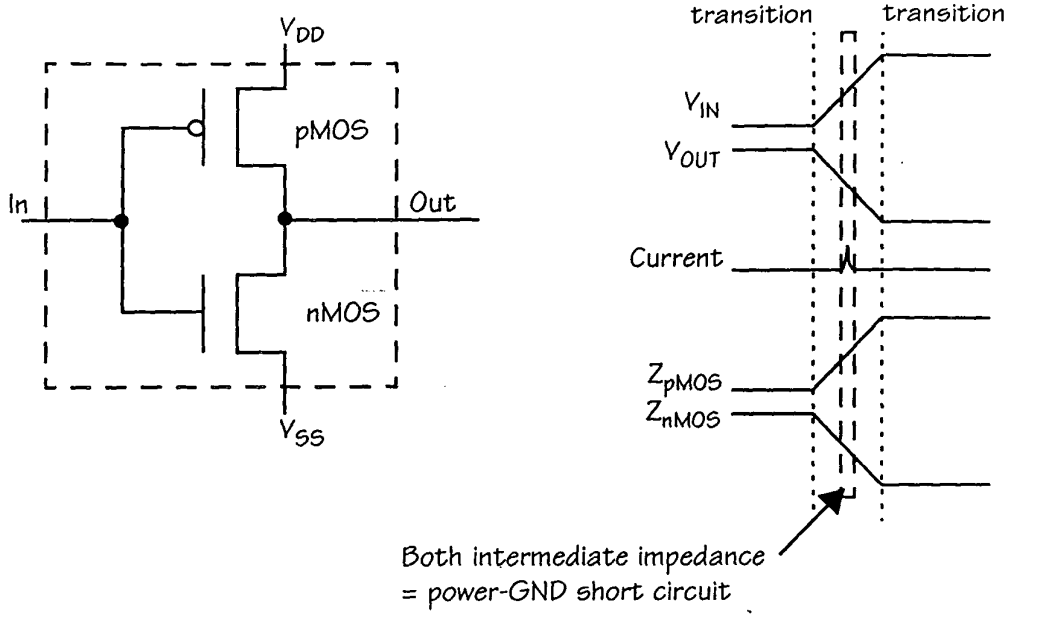


FIG. 179

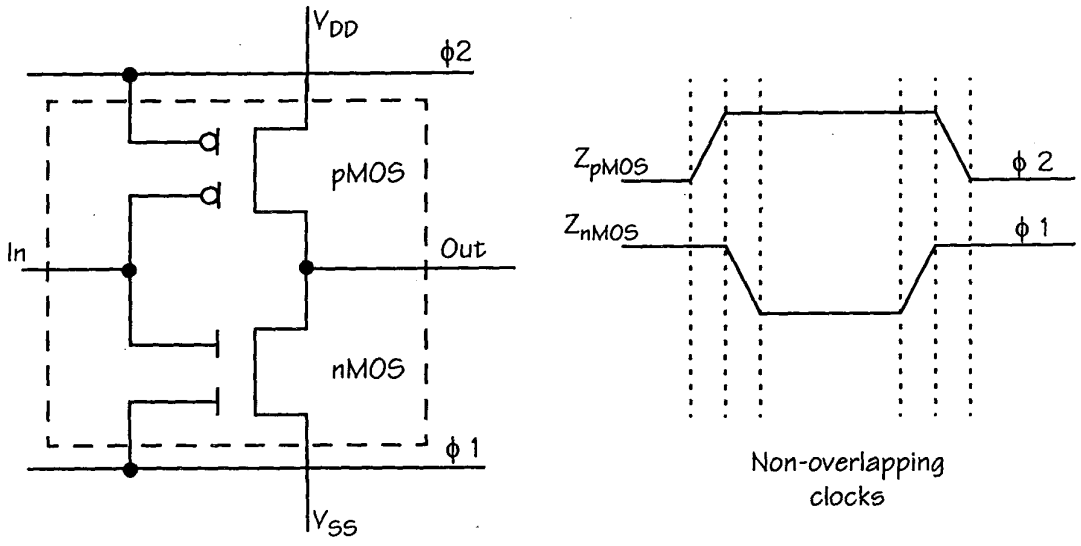


FIG. 180

Propeller Specification

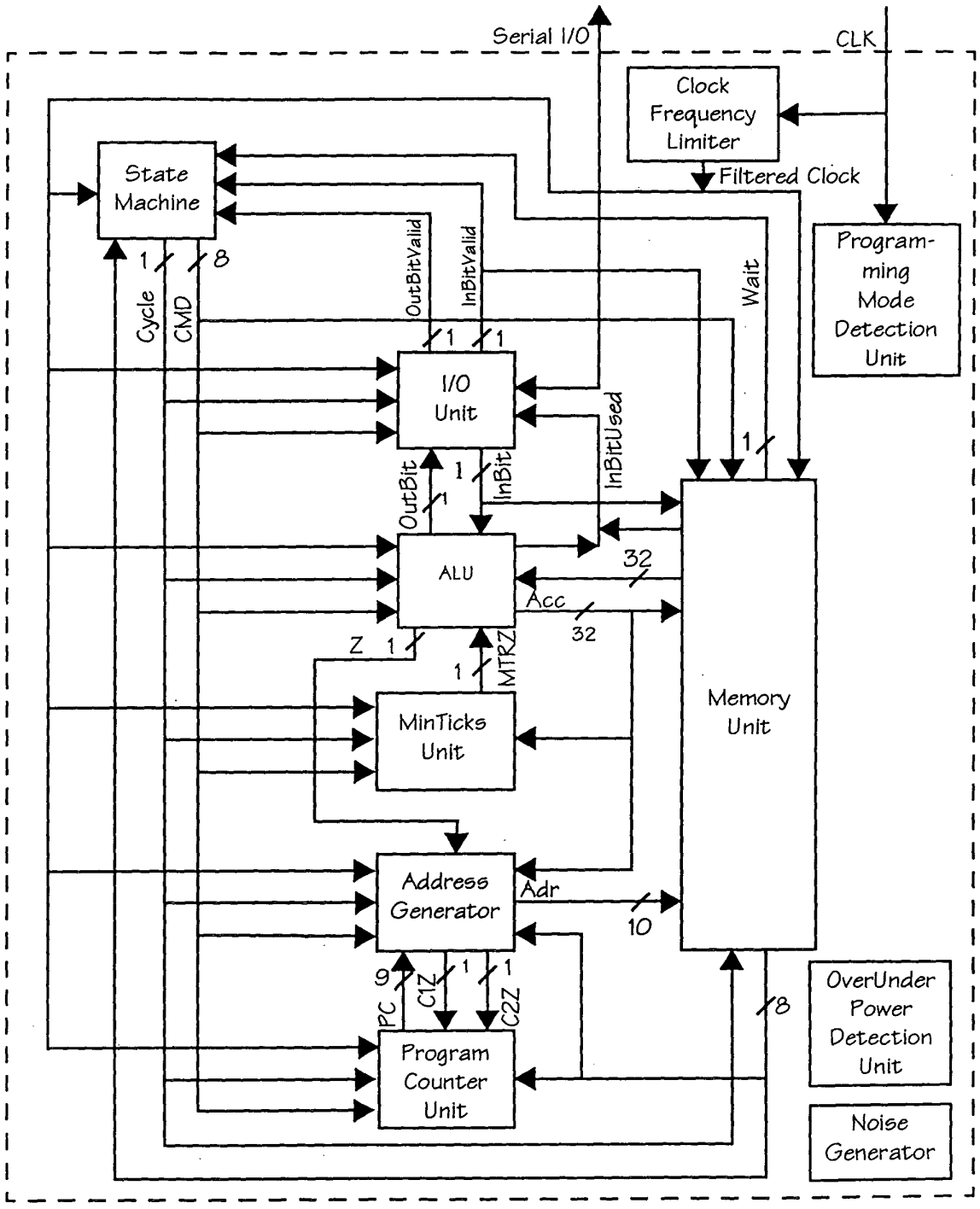


FIG. 181

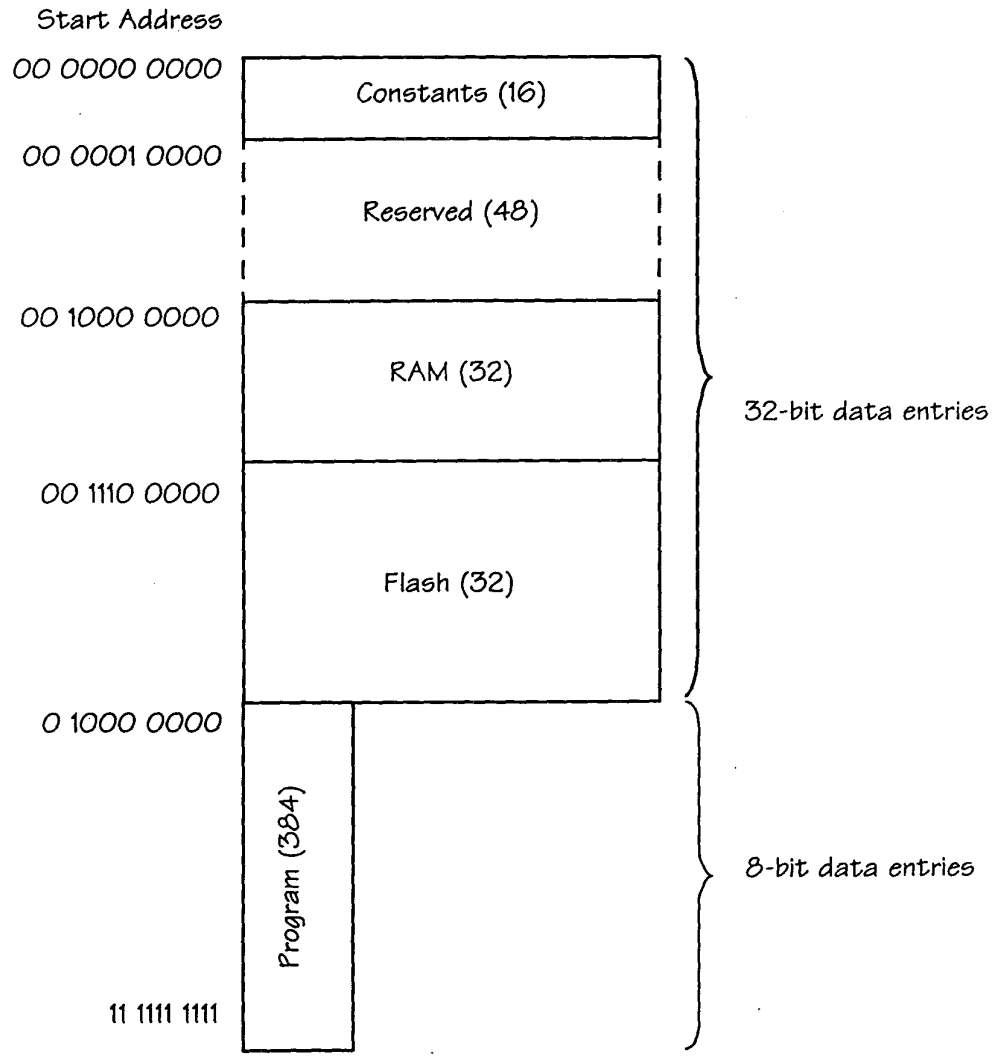

Start Address
00 0000 0000
Constants (16)
00 0001 0000
Reserved (48)
00 1000 0000
RAM (32)
32-bit data entries
00 1110 0000
Flash (32)
0 1000 0000
Program (384)
8-bit data entries
11 1111 1111

FIG. 182

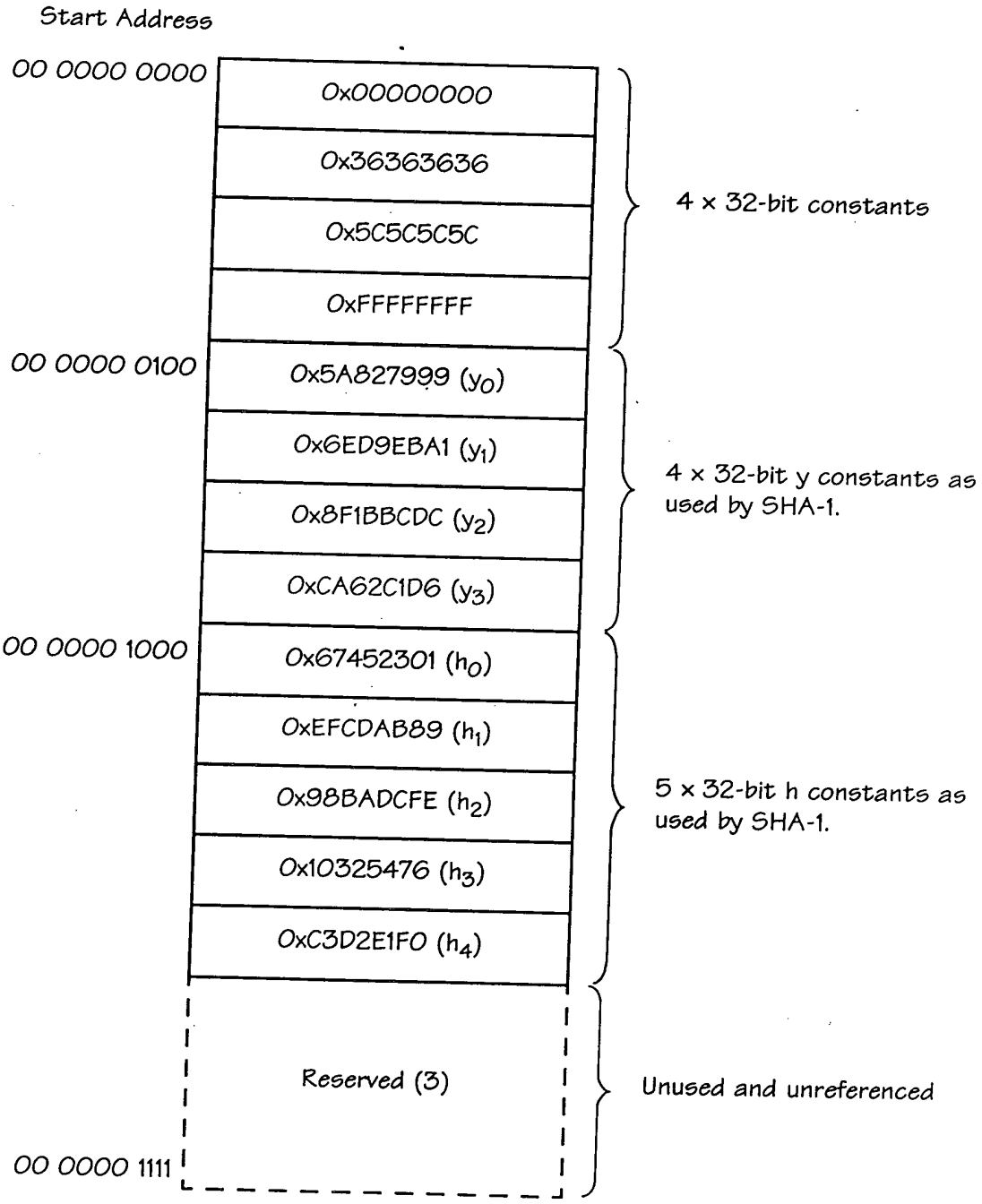

Start Address
00 0000 0000
0x00000000
0x36363636
0x5C5C5C5C
0xFFFFFFFF
00 0000 0100
0x5A827999 (y0)
0x6ED9EBA1 (y1)
0x8F1BBCDC (y2)
0xCA62C1D6 (y3)
00 0000 1000
0x67452301 (h0)
0xEFCDAB89 (h1)
0x98BADCFE (h2)
0x10325476 (h3)
0xC3D2E1F0 (h4)
Reserved (3)
00 0000 1111
4 x 32-bit constants
4 x 32-bit y constants as used by SHA-1.
5 x 32-bit h constants as used by SHA-1.
Unused and unreferenced

FIG. 183

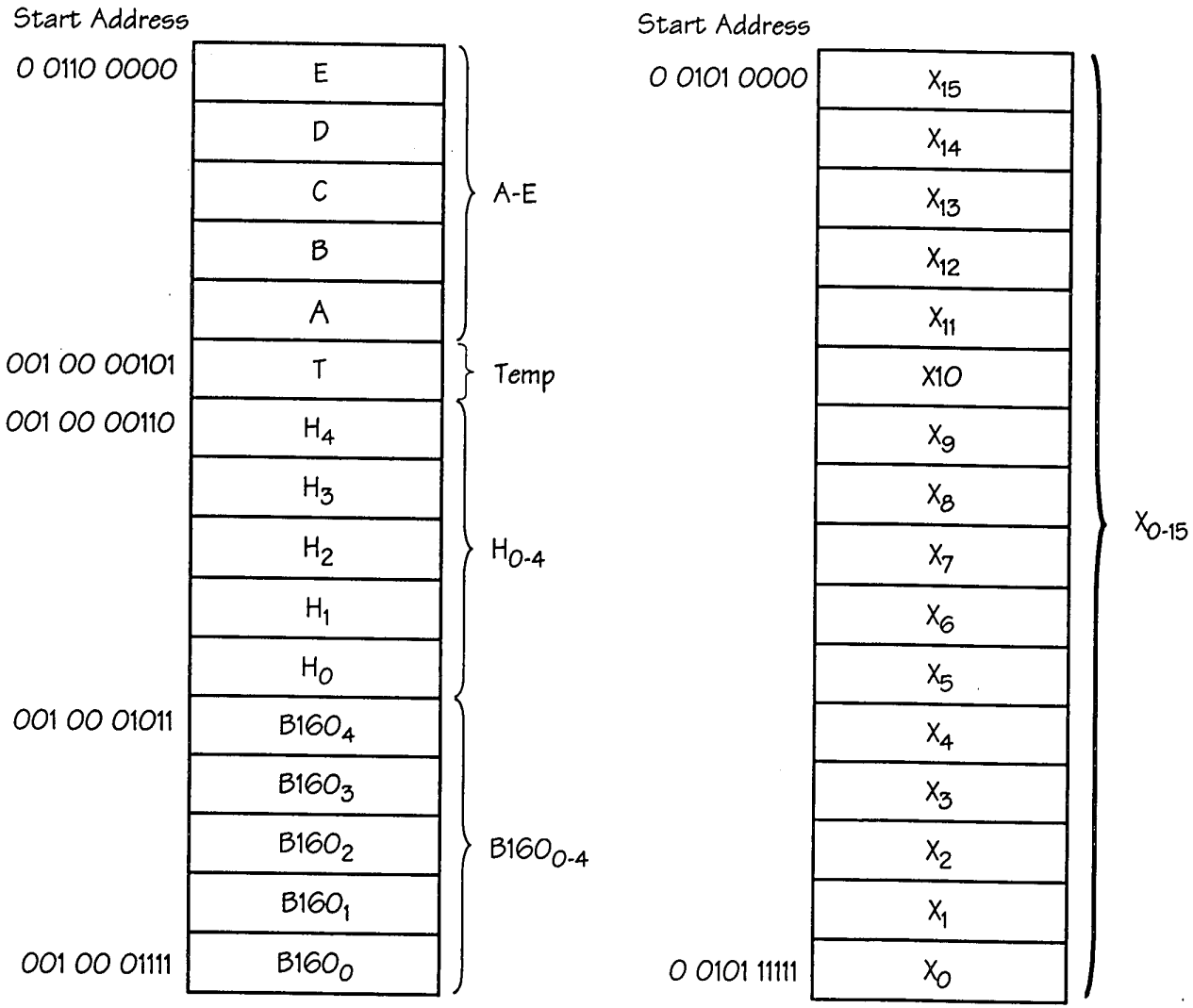

Start Address
0 0110 0000
E
D
C
B
A
A-E
001 00 00101
T
Temp
001 00 00110
H_4
H_3
H_2
H_1
H_0
H_{0-4}
001 00 01011
$B160_4$
$B160_3$
$B160_2$
$B160_1$
$B160_0$
$B160_{0-4}$
001 00 01111
Start Address
0 0101 0000
X_{15}
X_{14}
X_{13}
X_{12}
X_{11}
X10
X_9
X_8
X_7
X_6
X_5
X_4
X_3
X_2
X_1
X_0
X_{0-15}
0 0101 11111

FIG. 184

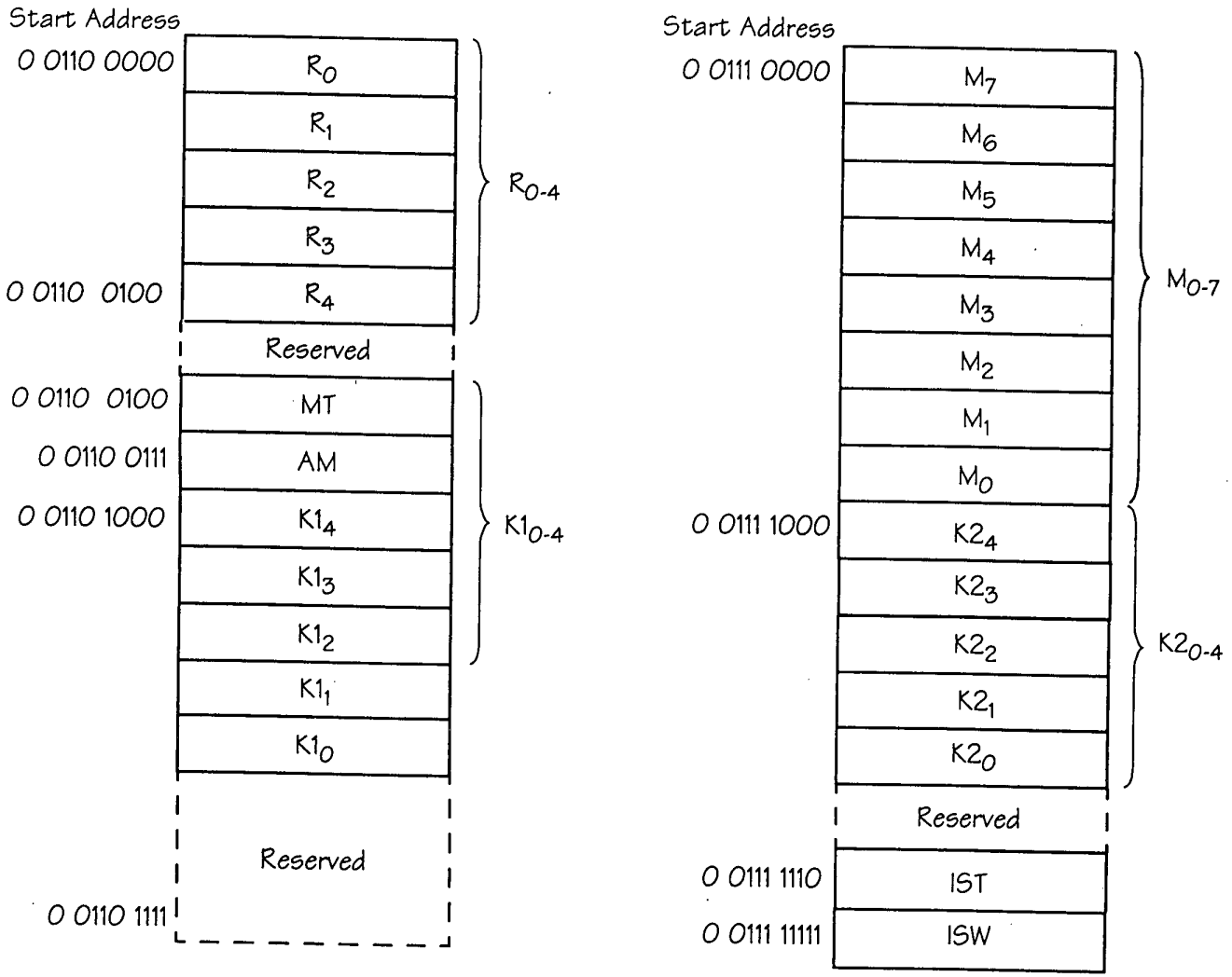

Start Address
0 0110 0000
R0
R1
R2
R3
0 0110 0100
R4
R0-4
Reserved
0 0110 0100
MT
0 0110 0111
AM
0 0110 1000
K1 4
K1 3
K1 2
K1 1
K1 0
K1 0-4
Reserved
0 0110 1111
Start Address
0 0111 0000
M7
M6
M5
M4
M3
M2
M1
M0
M0-7
0 0111 1000
K2 4
K2 3
K2 2
K2 1
K2 0
K2 0-4
Reserved
0 0111 1110
IST
0 0111 11111
ISW

FIG. 185

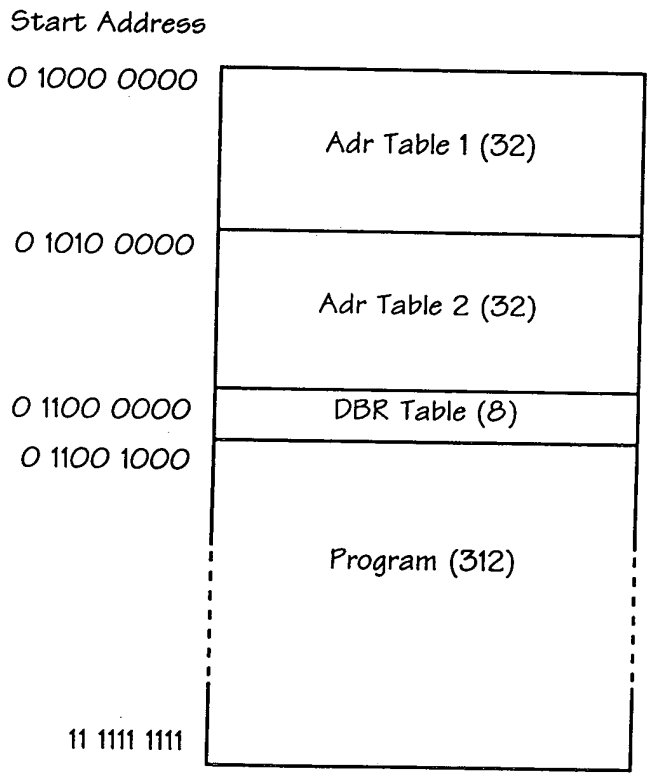

Start Address
0 1000 0000
Adr Table 1 (32)
0 1010 0000
Adr Table 2 (32)
0 1100 0000
DBR Table (8)
0 1100 1000
Program (312)
11 1111 1111

FIG. 186

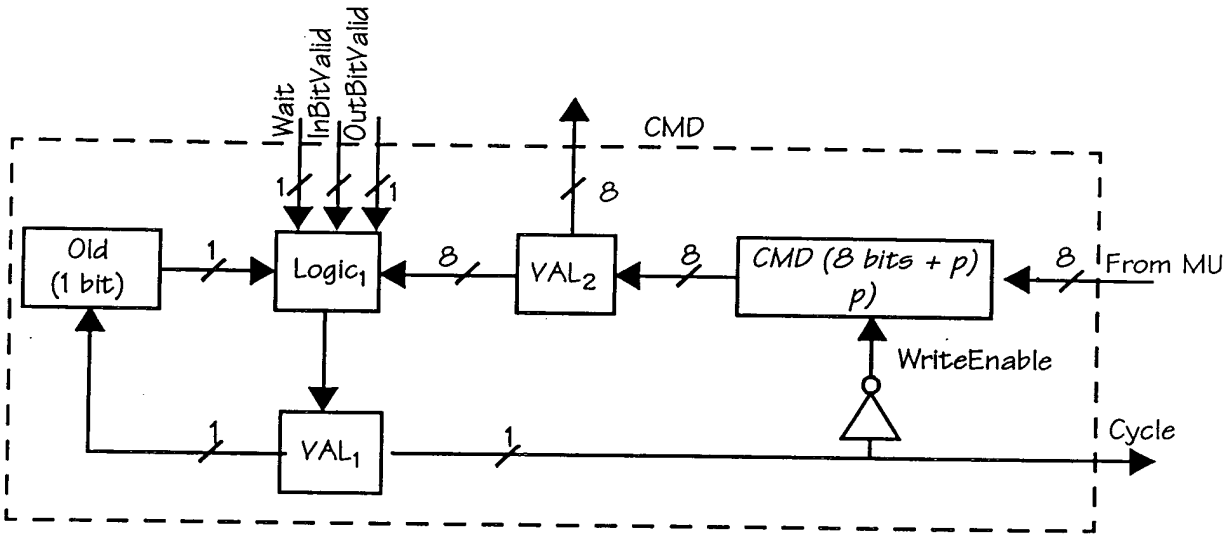


FIG. 187

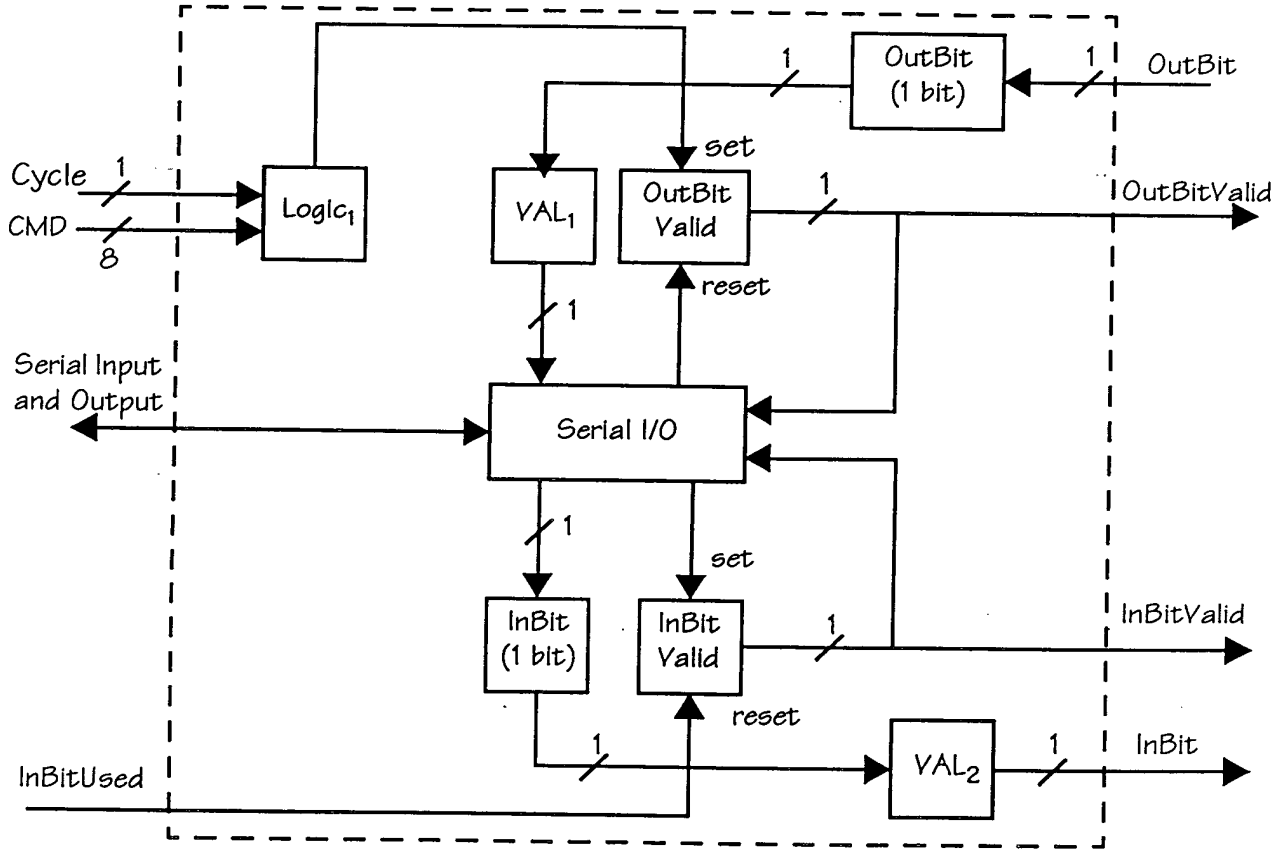


FIG. 188

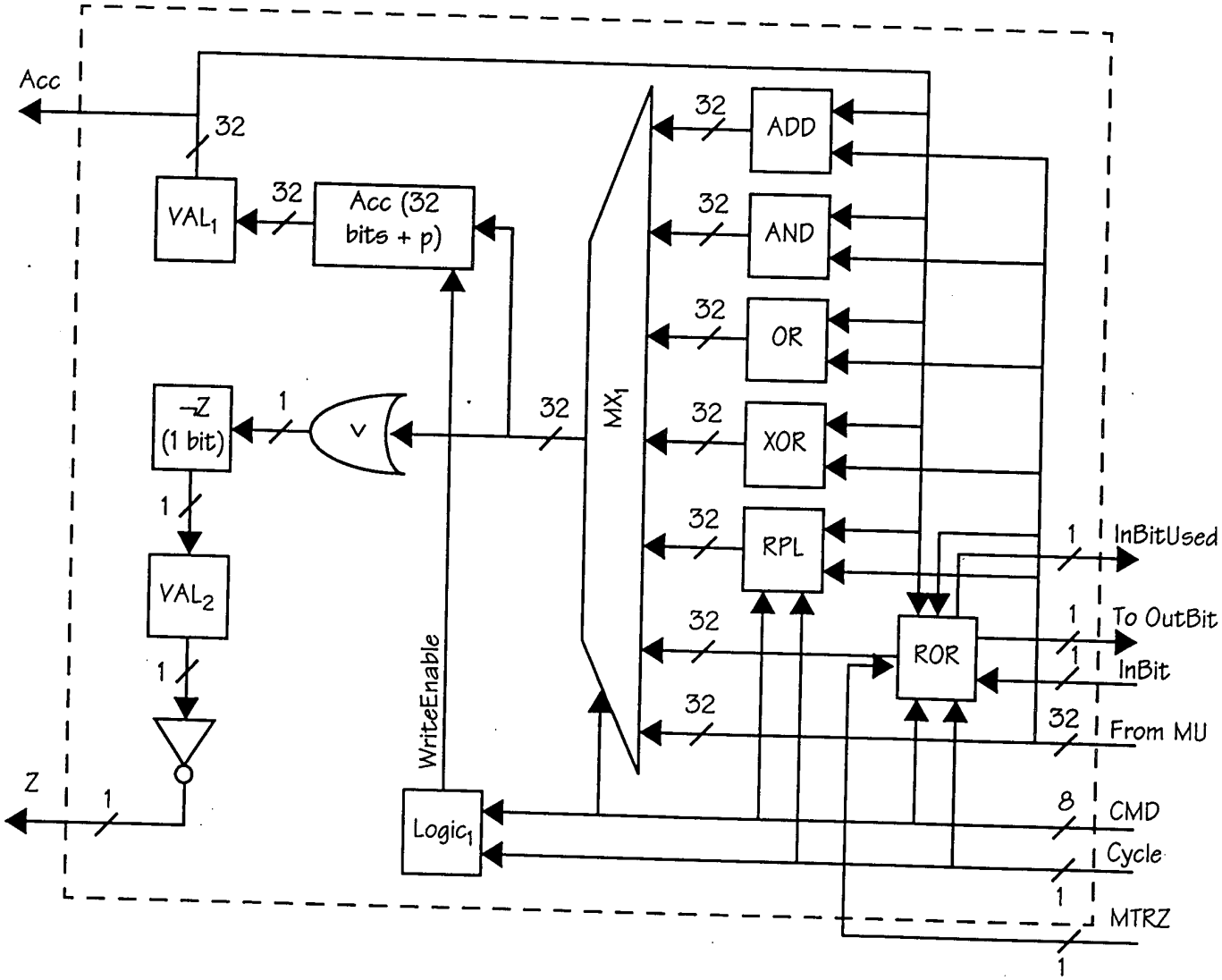

Acc
32
VAL1
32
Acc (32 bits + p)
32
ADD
32
AND
32
OR
32
XOR
32
RPL
32
ROR
32
From MU
32
MX1
32
WriteEnable
Logic1
¬Z (1 bit)
1
V
1
VAL2
1
Z
1
1
InBitUsed
1
To OutBit
1
InBit
8
CMD
Cycle
1
MTRZ
1

FIG. 189

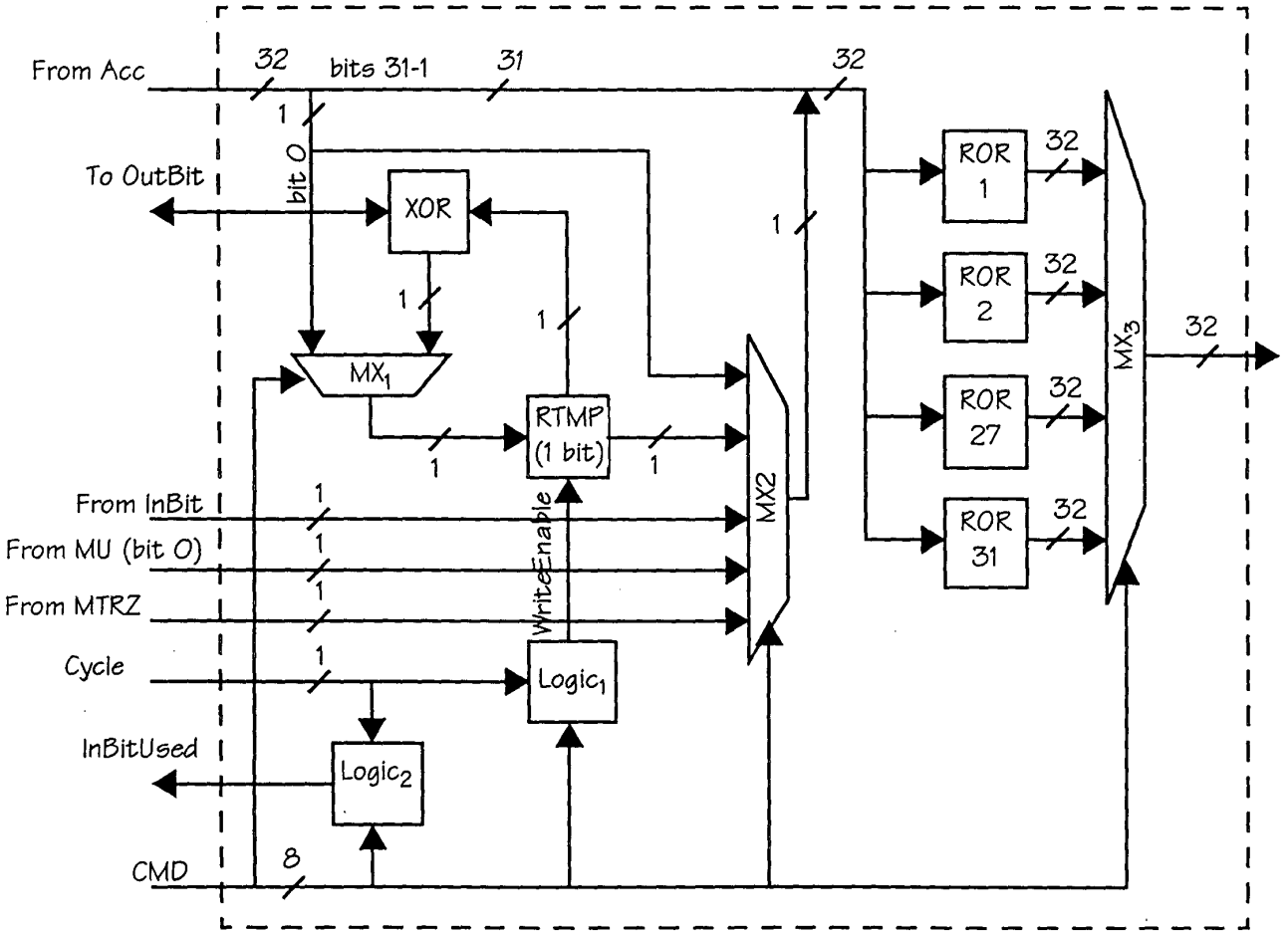

From Acc
32
bits 31-1
31
32
1
bit 0
To OutBit
XOR
1
1
1
MX1
RTMP
(1 bit)
1
1
From InBit
From MU (bit 0)
From MTRZ
Cycle
InBitUsed
CMD
8
1
1
1
1
WriteEnable
Logic1
Logic2
MX2
ROR
1
ROR
2
ROR
27
ROR
31
32
32
32
32
MX3
32

FIG. 190

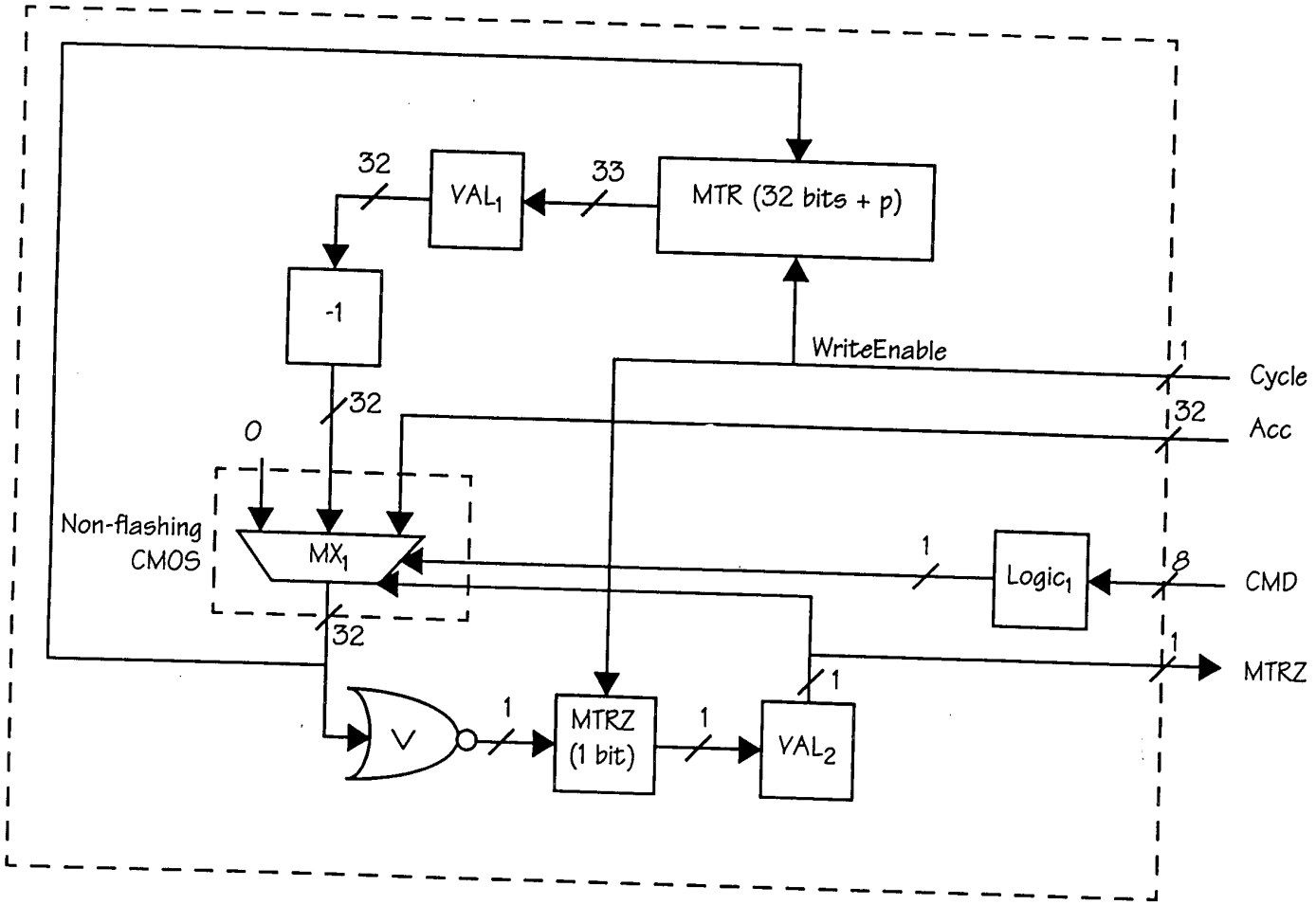

32
VAL1
33
MTR (32 bits + p)
-1
WriteEnable
1
Cycle
32
Acc
32
0
Non-flashing
CMOS
MX1
1
Logic1
8
CMD
32
1
MTRZ
V
1
MTRZ
(1 bit)
1
1
VAL2

FIG. 191

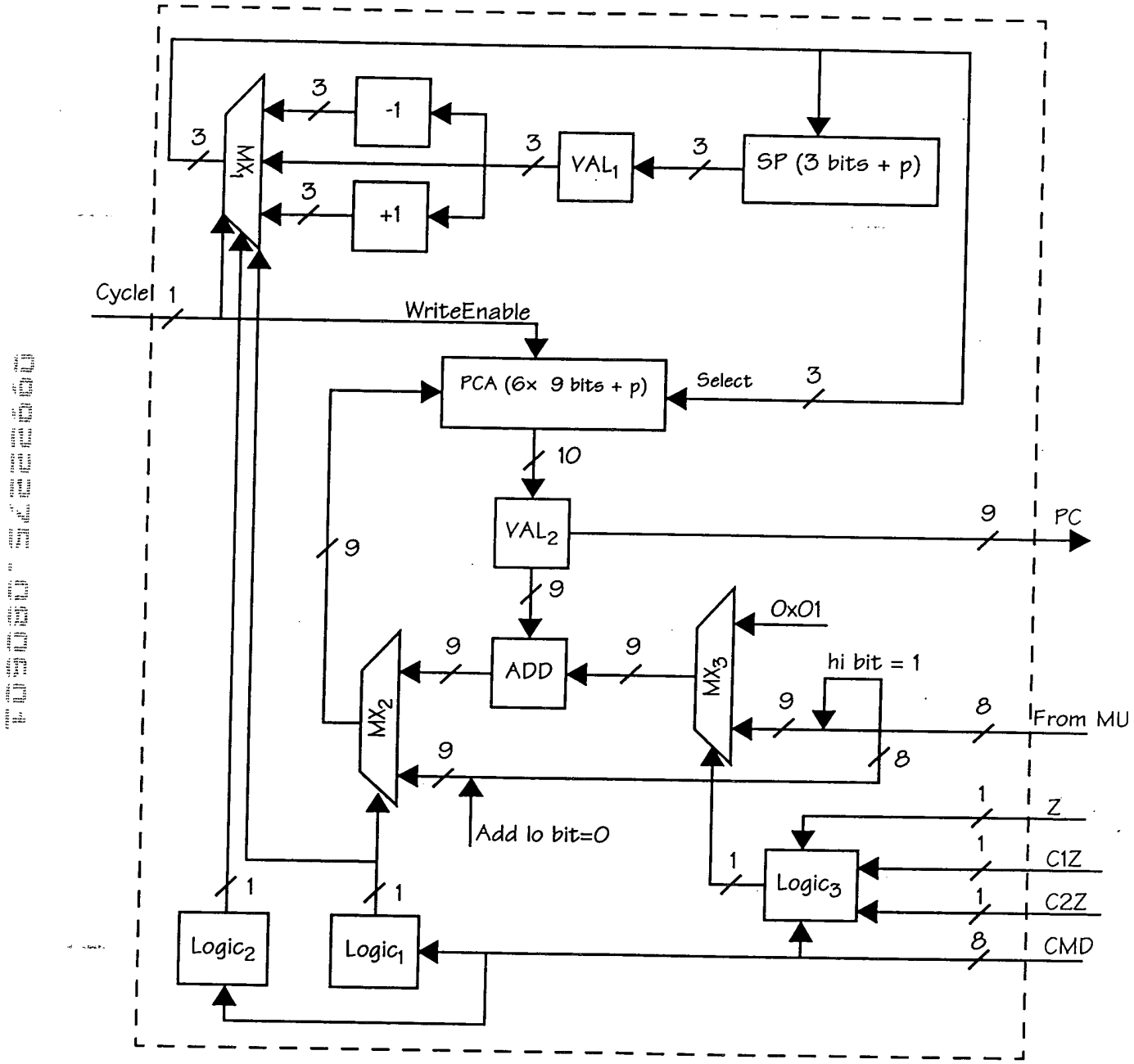

3
-1
3
MX1
3
VAL1
3
SP (3 bits + p)
3
+1
Cycle
1
WriteEnable
PCA (6x 9 bits + p)
Select
3
10
VAL2
9
PC
9
9
OxO1
ADD
9
9
MX3
hi bit = 1
MX2
9
8
From MU
9
8
1
Z
Add lo bit=0
1
C1Z
1
Logic3
1
C2Z
1
1
8
CMD
Logic2
Logic1

FIG. 192

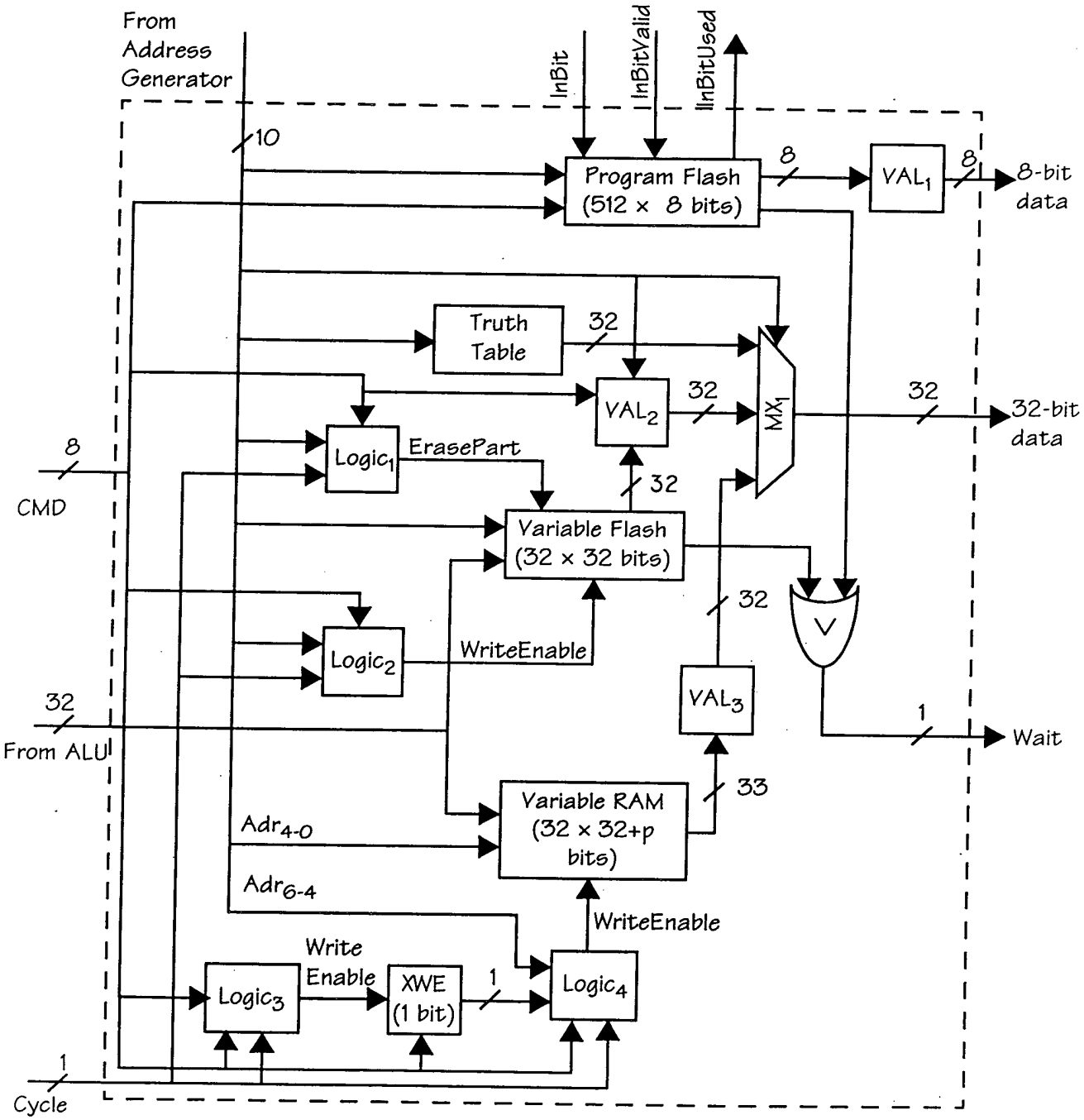

From Address Generator
10
InBit
InBitValid
InBitUsed
Program Flash (512 × 8 bits)
8
VAL1
8-bit data
Truth Table
32
VAL2
MX1
32-bit data
8
CMD
Logic1
ErasePart
Variable Flash (32 × 32 bits)
Logic2
WriteEnable
V
VAL3
32
From ALU
Wait
Variable RAM (32 × 32+p bits)
33
Adr4-0
Adr6-4
WriteEnable
Write Enable
Logic3
XWE (1 bit)
Logic4
Cycle

FIG. 193

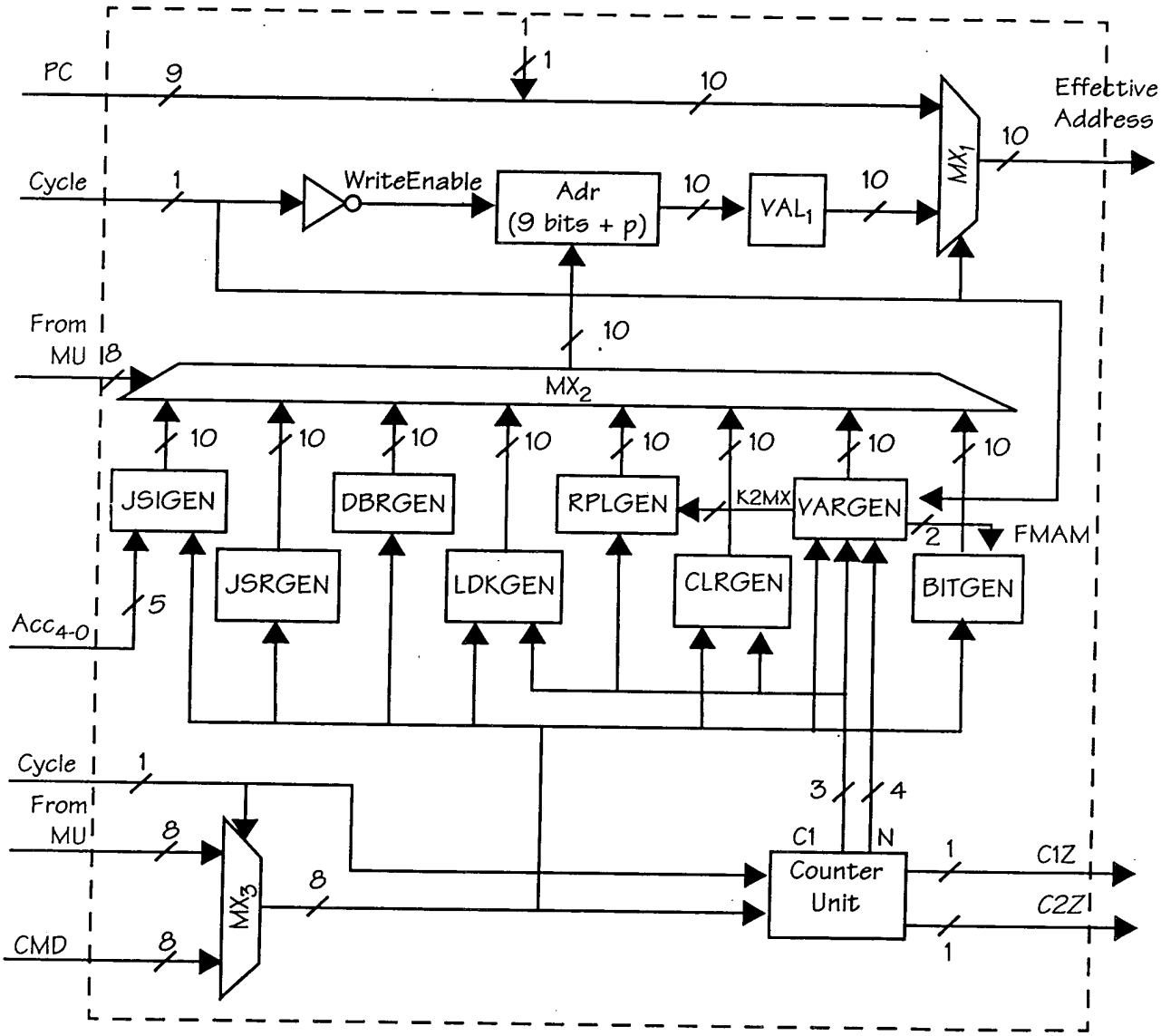

PC
9
1
1
10
Effective
Address
10
MX1
Cycle
1
WriteEnable
Adr
(9 bits + p)
10
VAL1
10
From
MU
8
10
MX2
10
10
10
10
10
10
10
10
JSIGEN
DBRGEN
RPLGEN
K2MX
VARGEN
2
FMAM
JSRGEN
LDKGEN
CLRGEN
BITGEN
Acc4-0
5
Cycle
1
From
MU
8
CMD
8
MX3
8
3
4
C1
N
Counter
Unit
1
C1Z
C2Z
1

FIG. 194

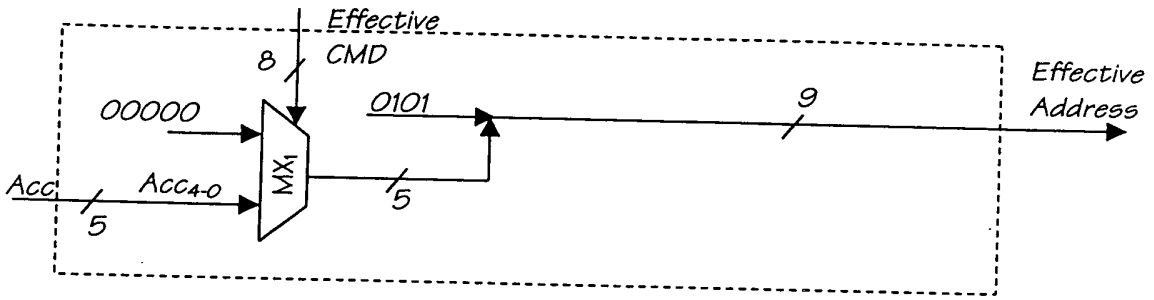


FIG. 195

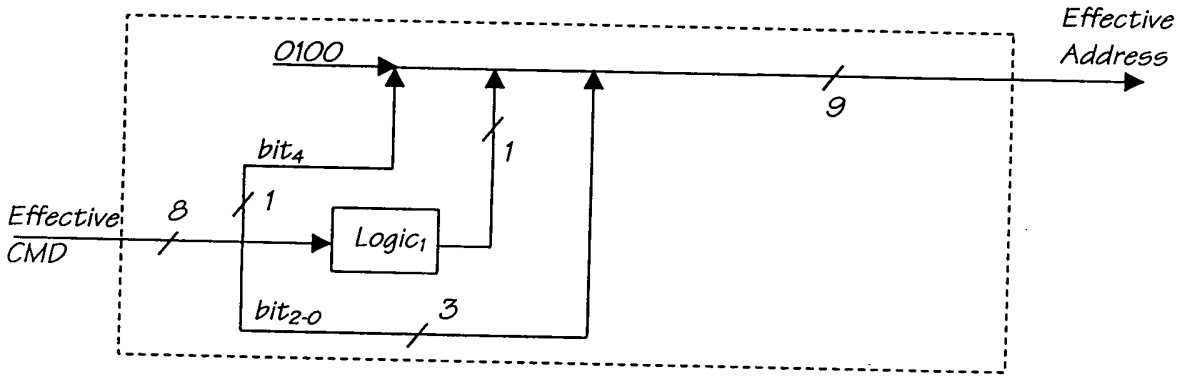


FIG. 196

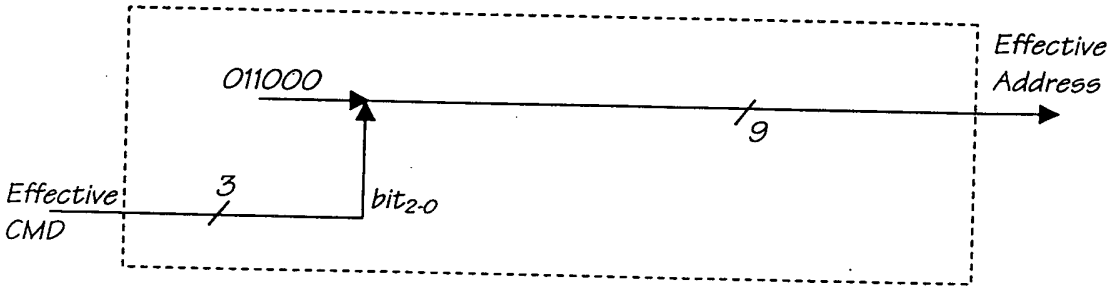


FIG. 197

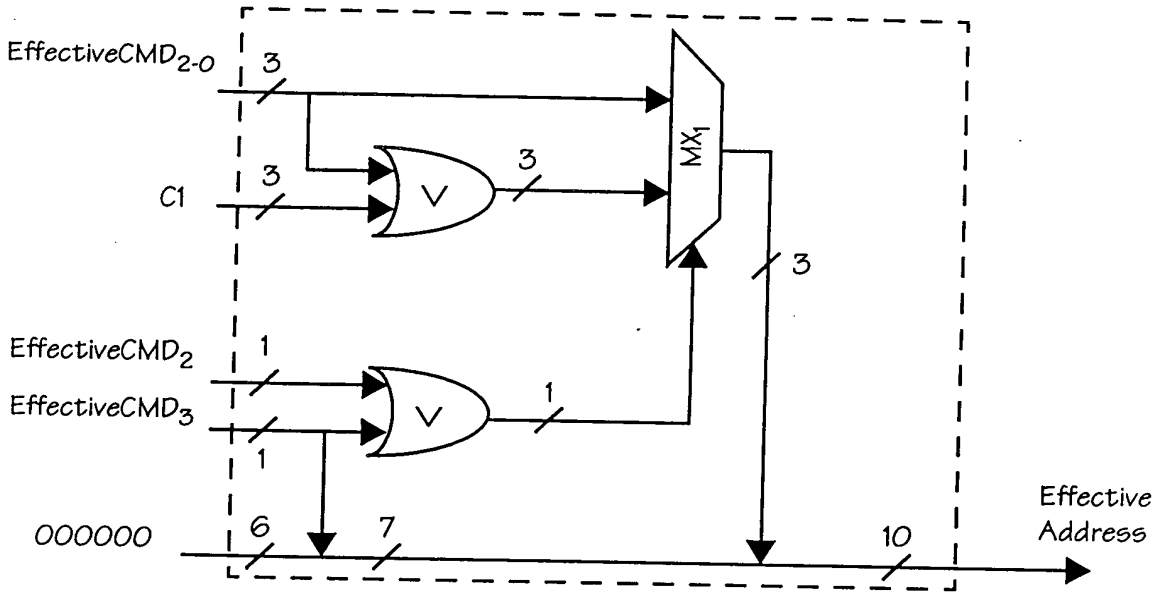


FIG. 198

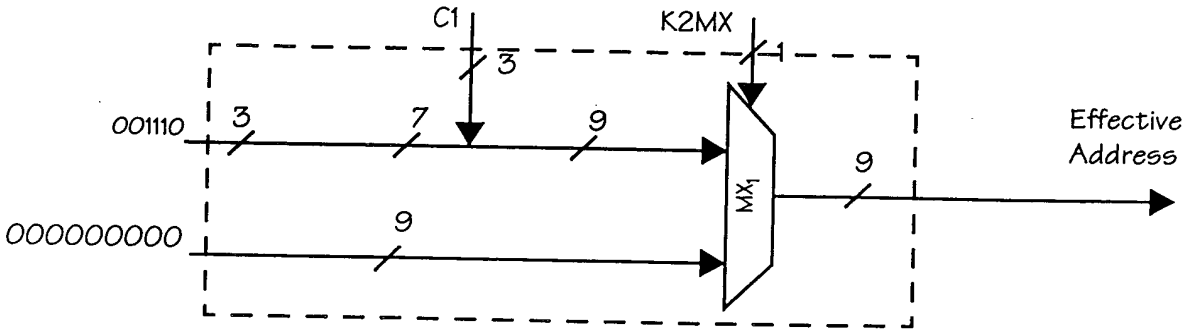


FIG. 199

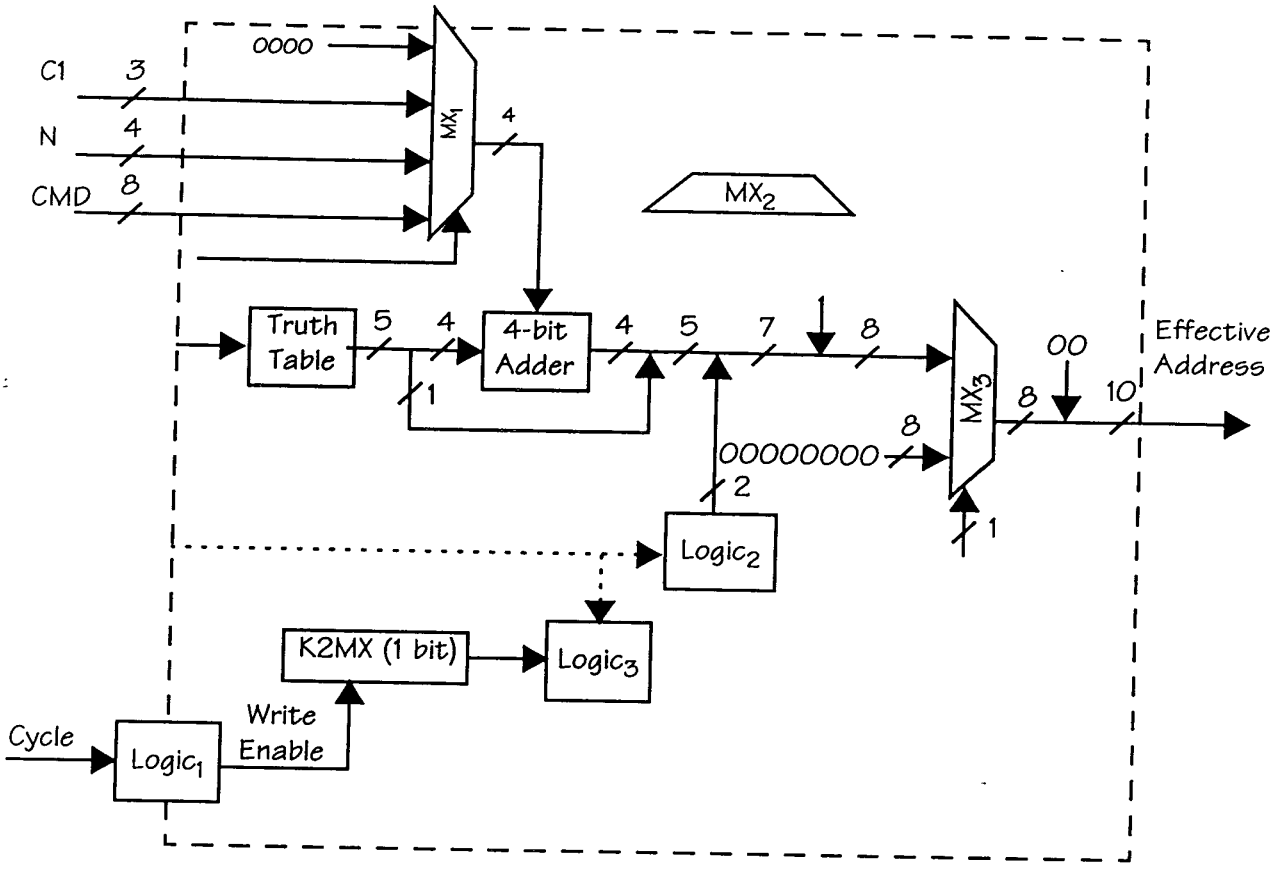


FIG. 200

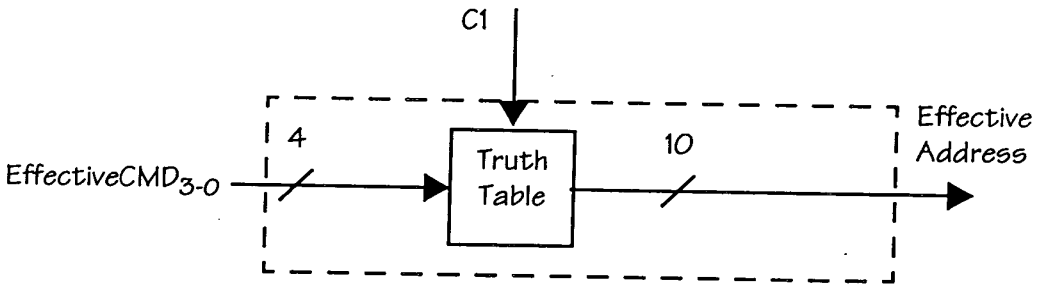


FIG. 201

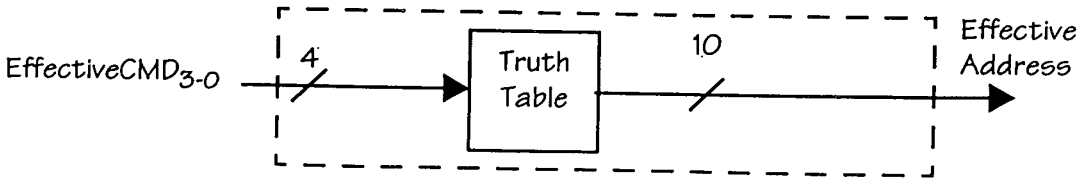


FIG. 202

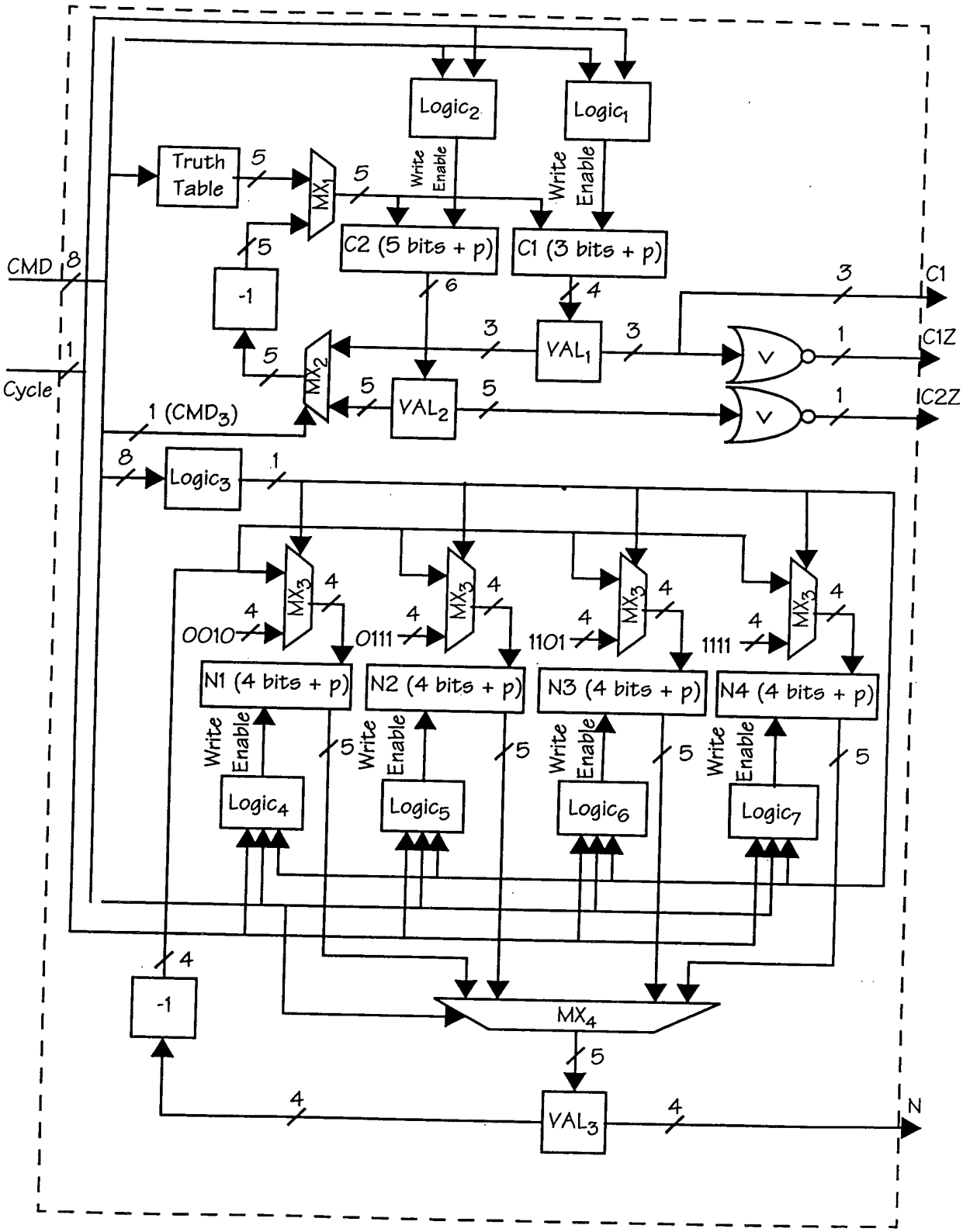

Logic2
Logic1
Truth Table
5
MX1
5
Write Enable
Write Enable
C2 (5 bits + p)
C1 (3 bits + p)
CMD
8
-1
5
6
4
3
C1
3
VAL1
3
V
1
C1Z
Cycle
1
MX2
5
5
VAL2
5
V
1
IC2Z
1 (CMD3)
8
Logic3
1
MX3
4
4
0010
4
0111
4
1101
4
1111
N1 (4 bits + p)
N2 (4 bits + p)
N3 (4 bits + p)
N4 (4 bits + p)
Write Enable
5
Logic4
Logic5
Logic6
Logic7
4
-1
MX4
5
4
VAL3
4
N

FIG. 203

705

DATA TYPE	BITS
Factory Code	16
Batch Number	32
Serial Number	48
Manufacturing Date	16
Media Length	24
Media Type	8
Preprinted Media Length	16
Cyan Ink Viscosity	8
Magenta Ink Viscosity	8
Yellow Ink Viscosity	8
Cyan Drop Volume	8
Magenta Drop Volume	8
Yellow Drop Volume	8
Cyan Ink Color	24
Magenta Ink Color	24
Yellow Ink Color	24
Remaining-media Length Indicator	16
Authentication Key	128
Copyrightable bit pattern	512
Reserved for Camera Use	88
TOTAL	1024

728

FIG. 204

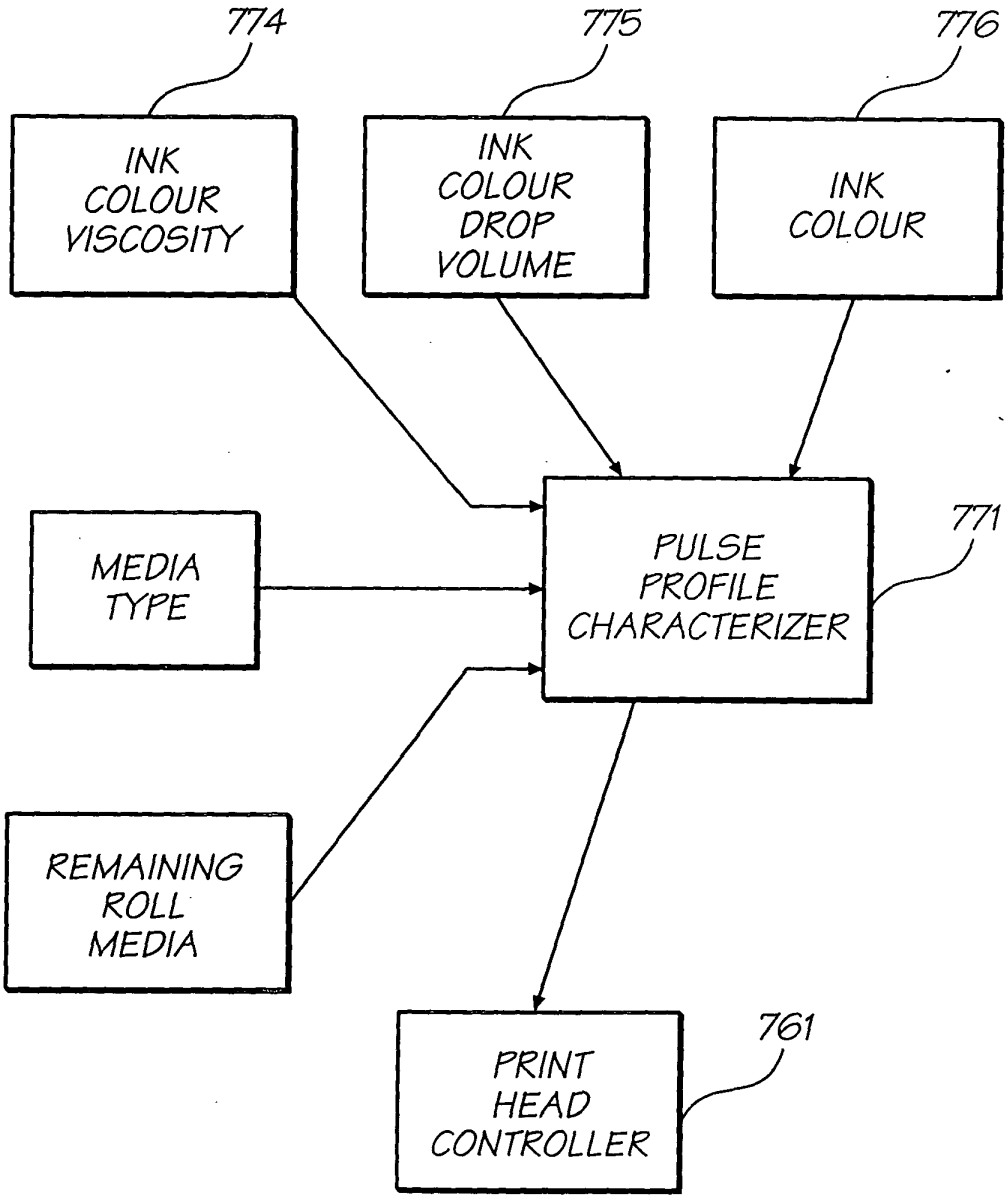

774
775
776
INK COLOUR VISCOSITY
INK COLOUR DROP VOLUME
INK COLOUR
MEDIA TYPE
PULSE PROFILE CHARACTERIZER
771
REMAINING ROLL MEDIA
761
PRINT HEAD CONTROLLER

FIG. 205

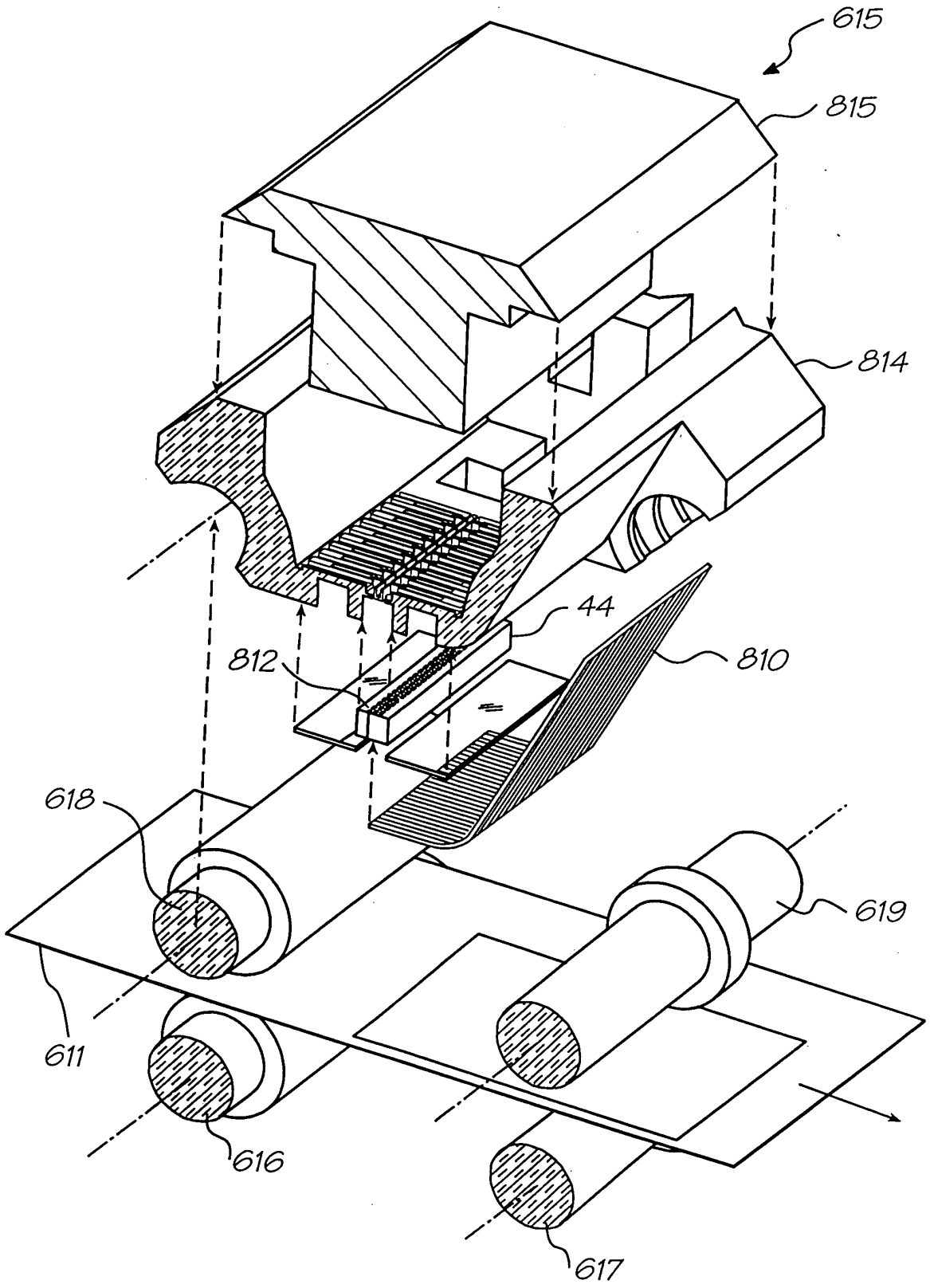


FIG. 206

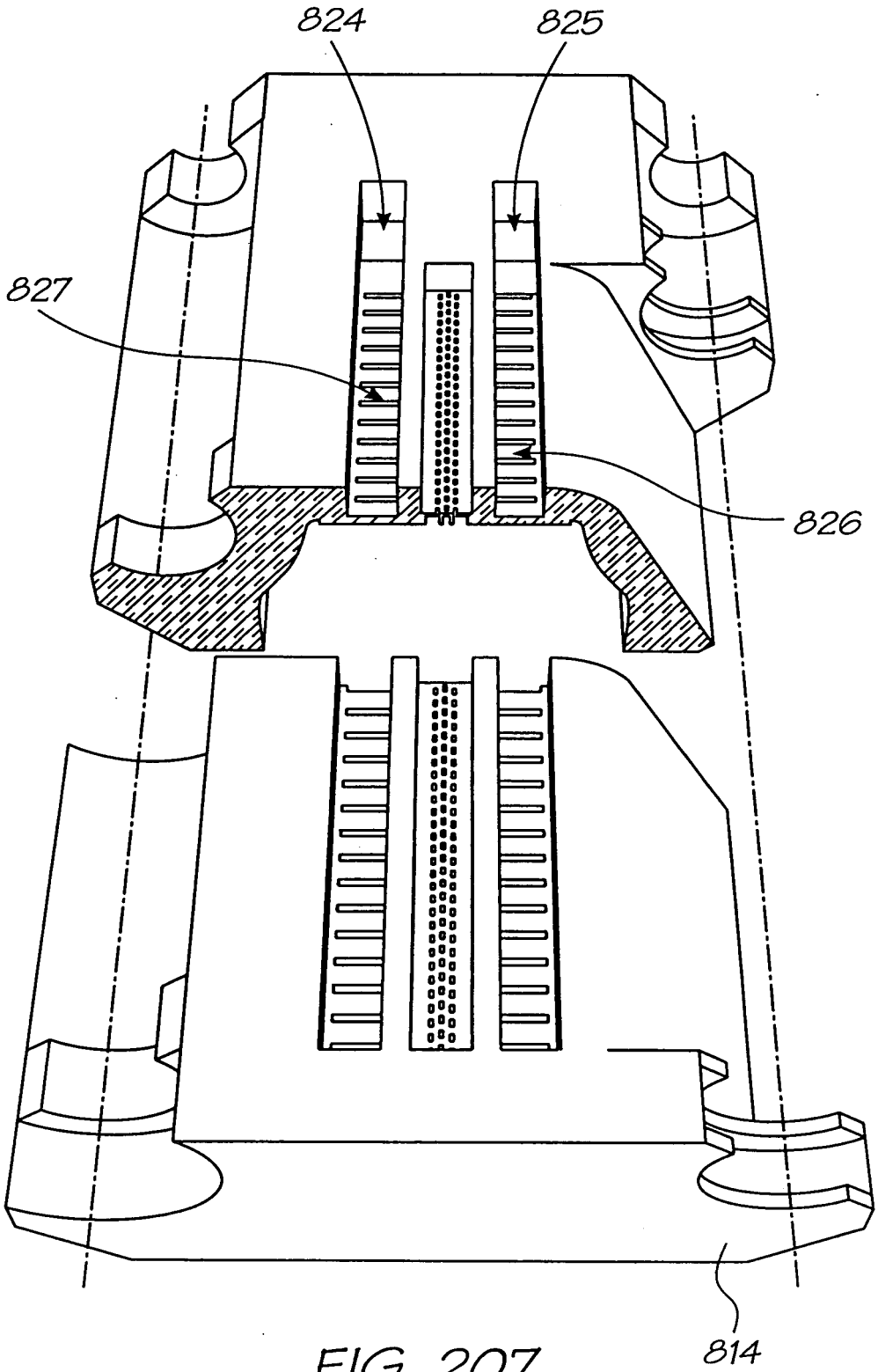


FIG. 207

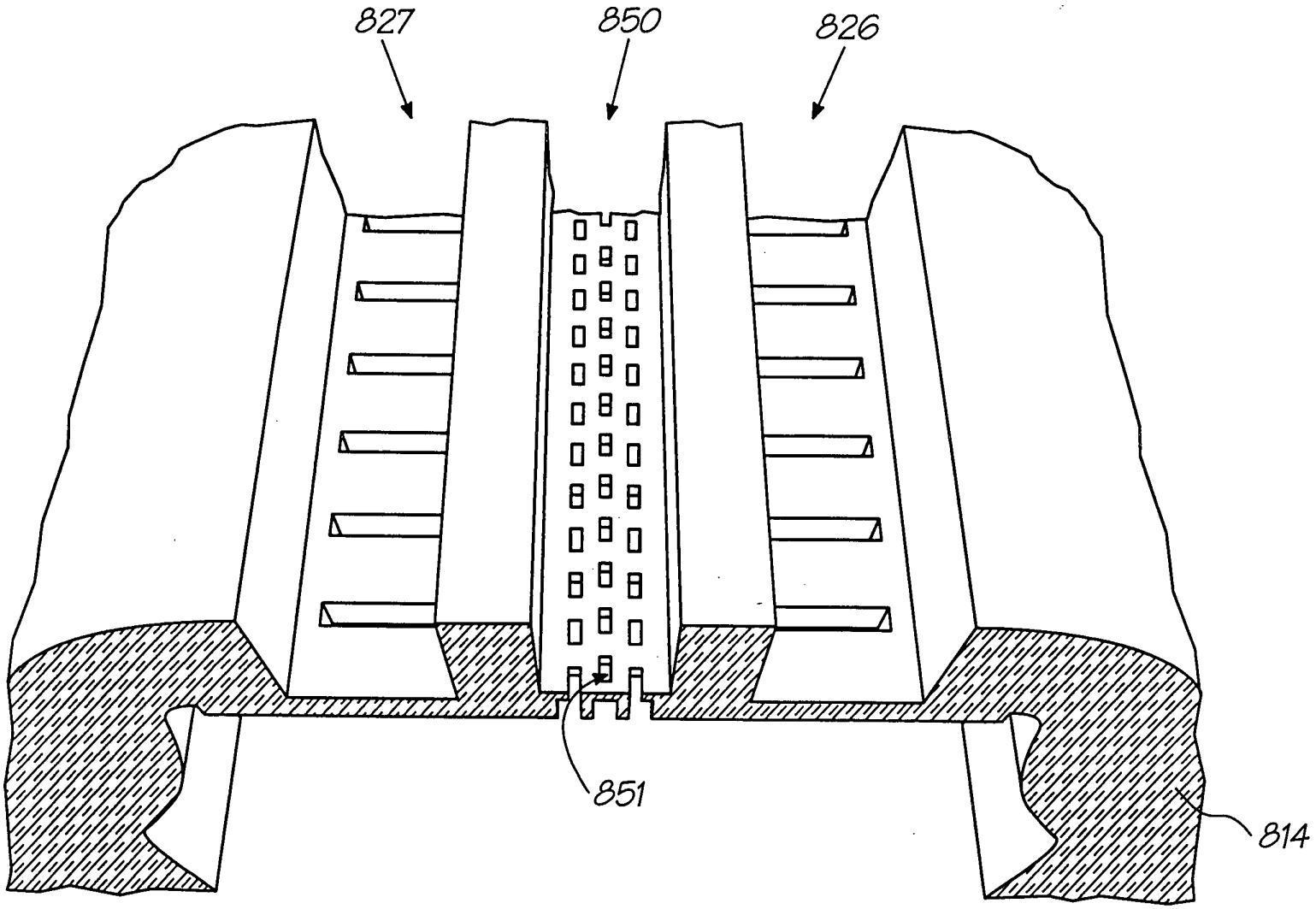


FIG. 208

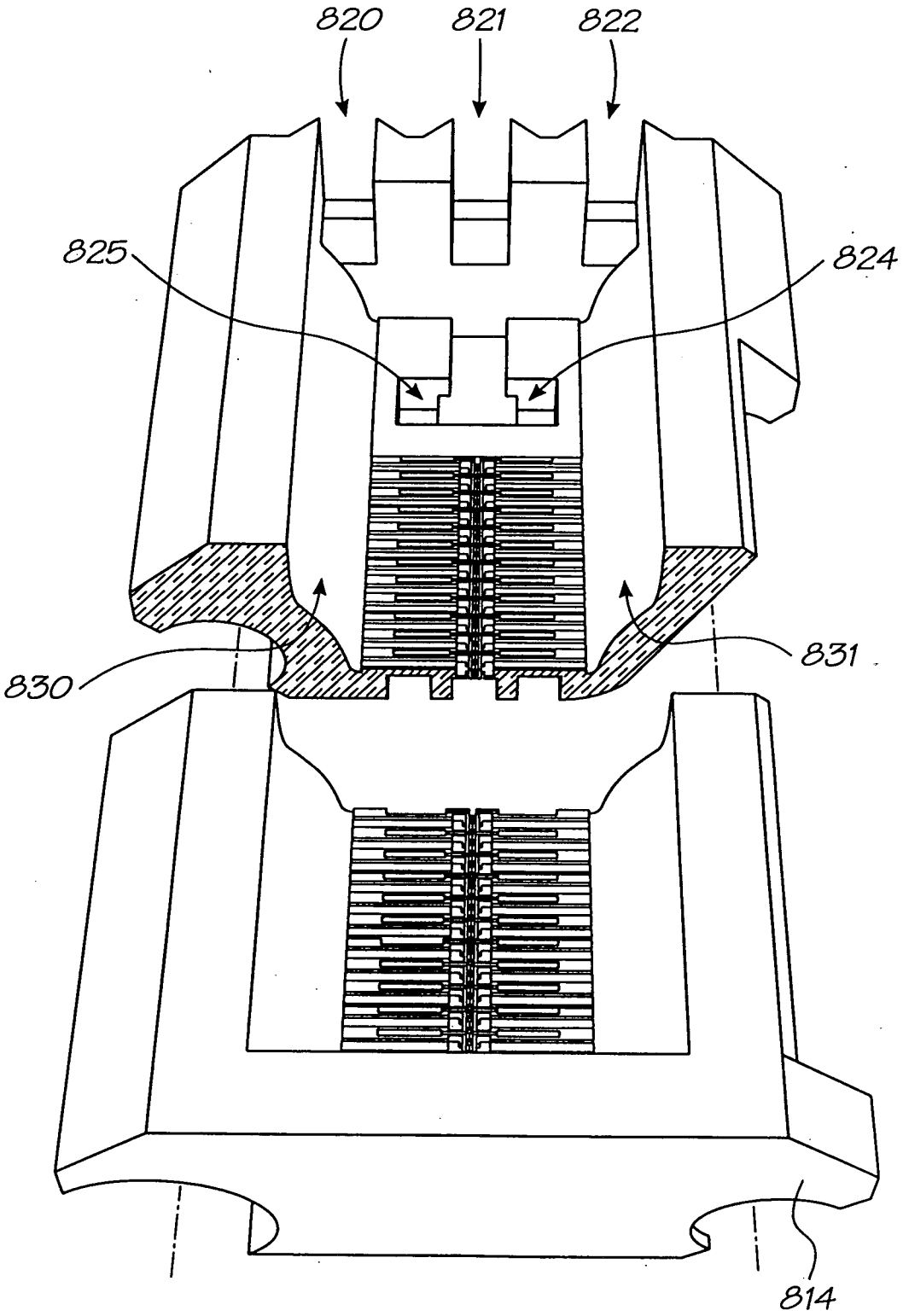


FIG. 209

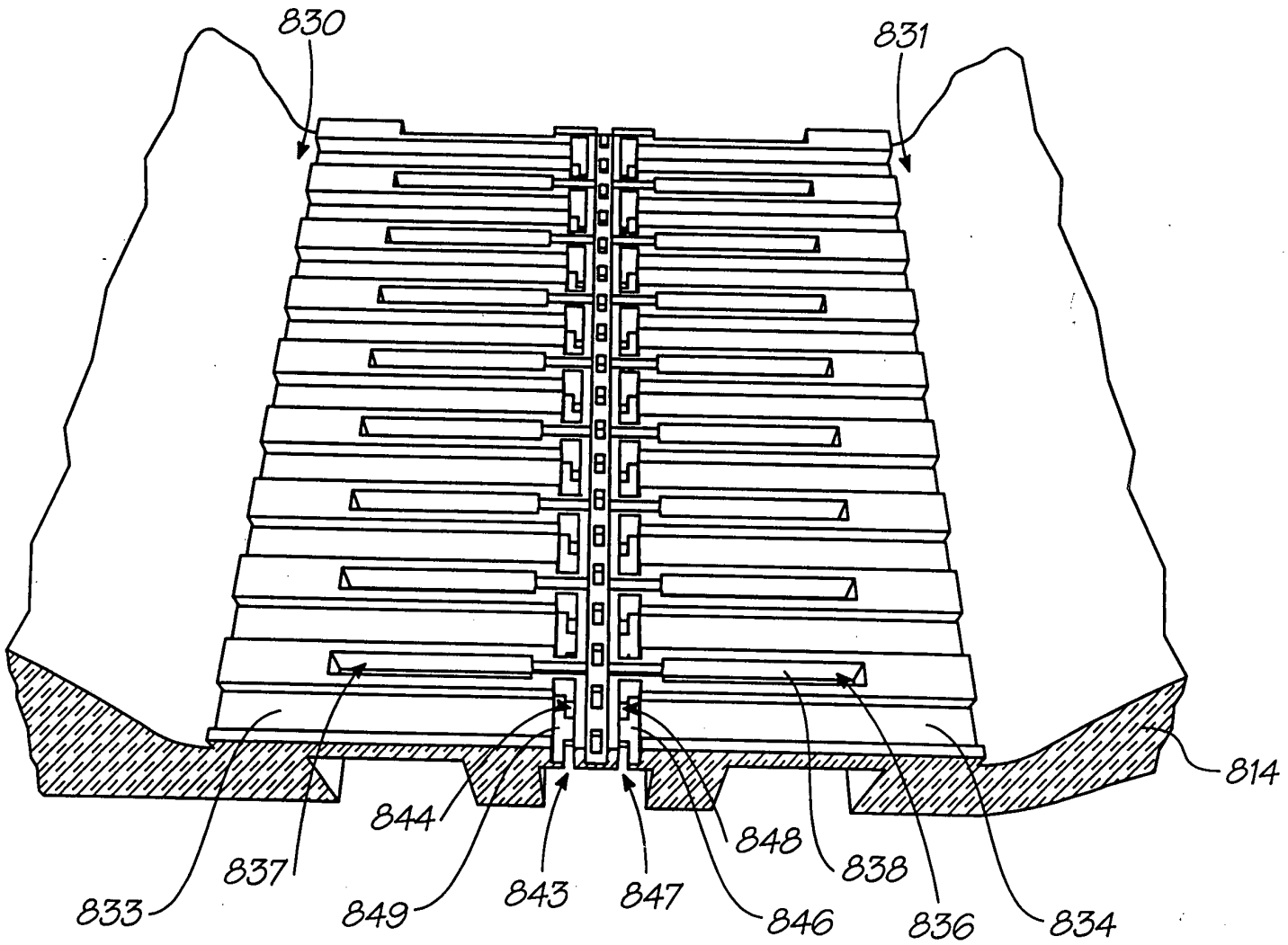


FIG. 210

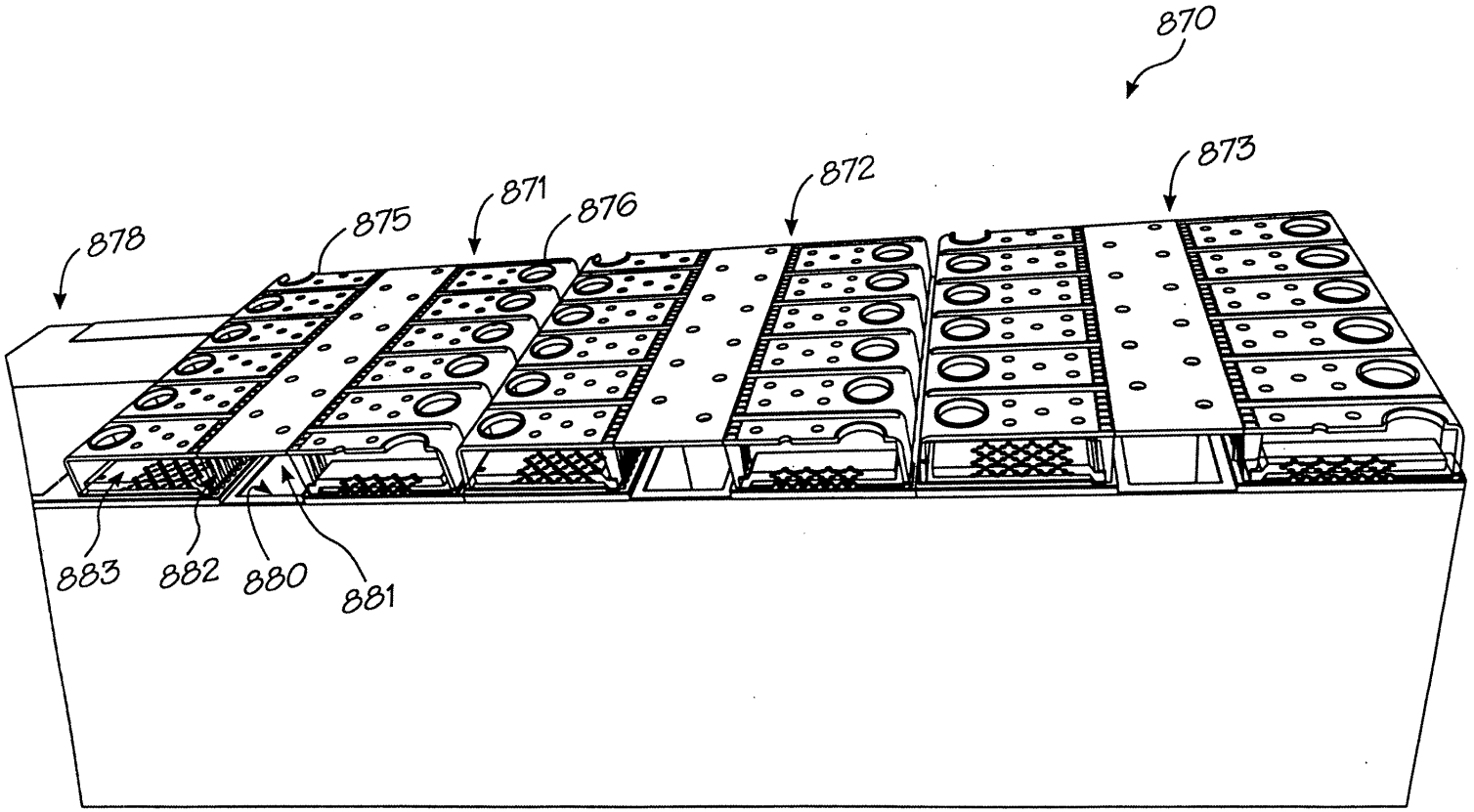


FIG. 211

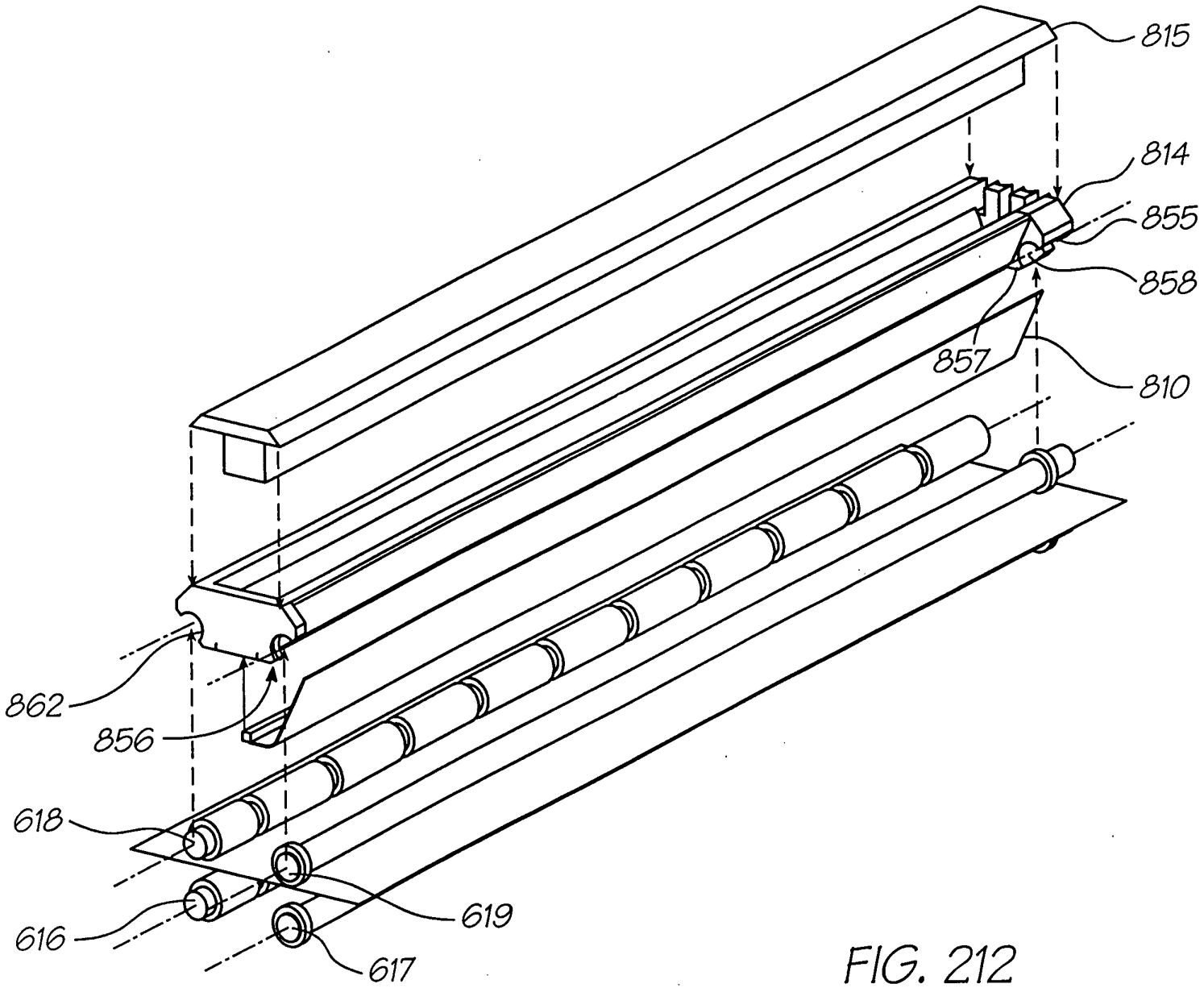


FIG. 212

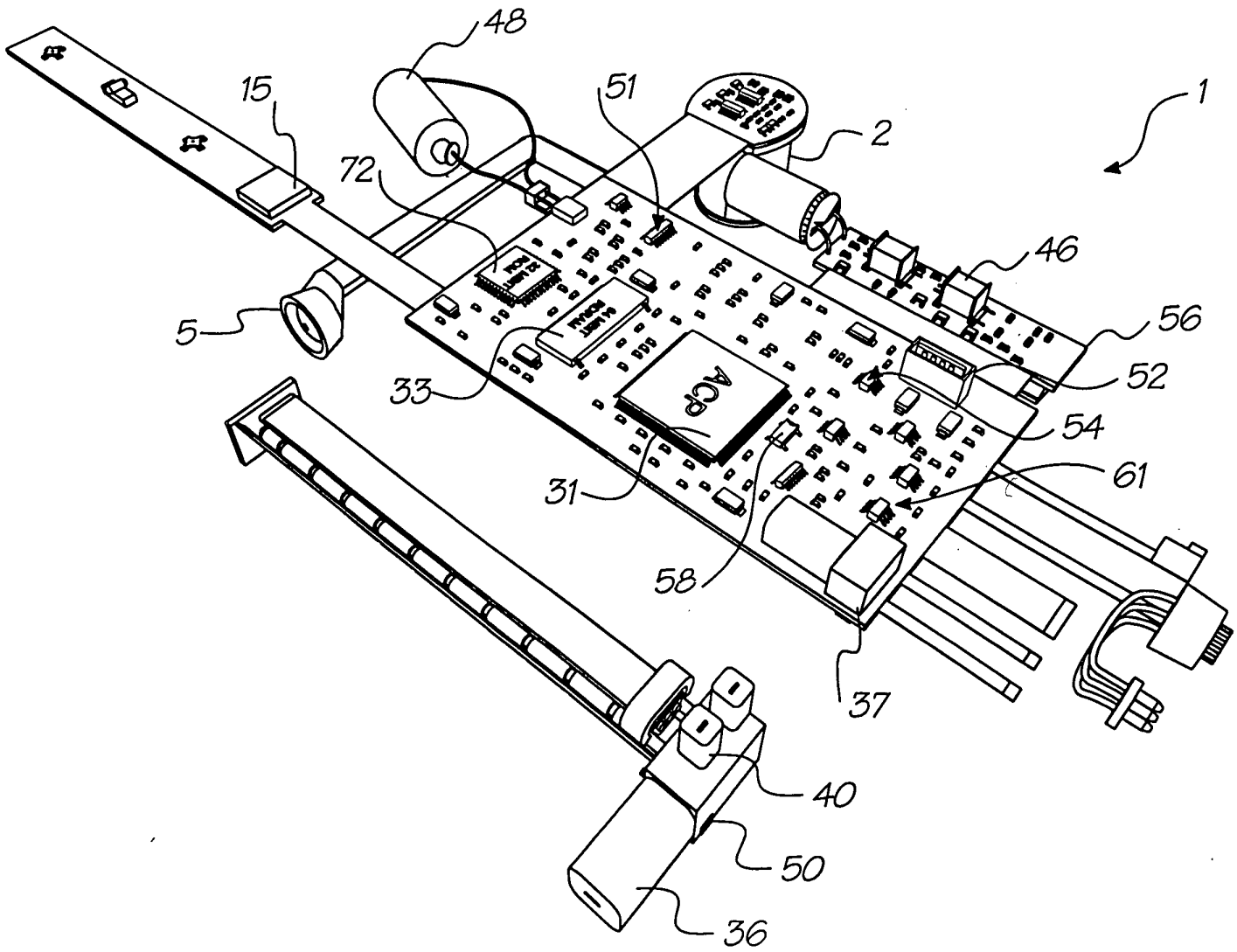


FIG. 213

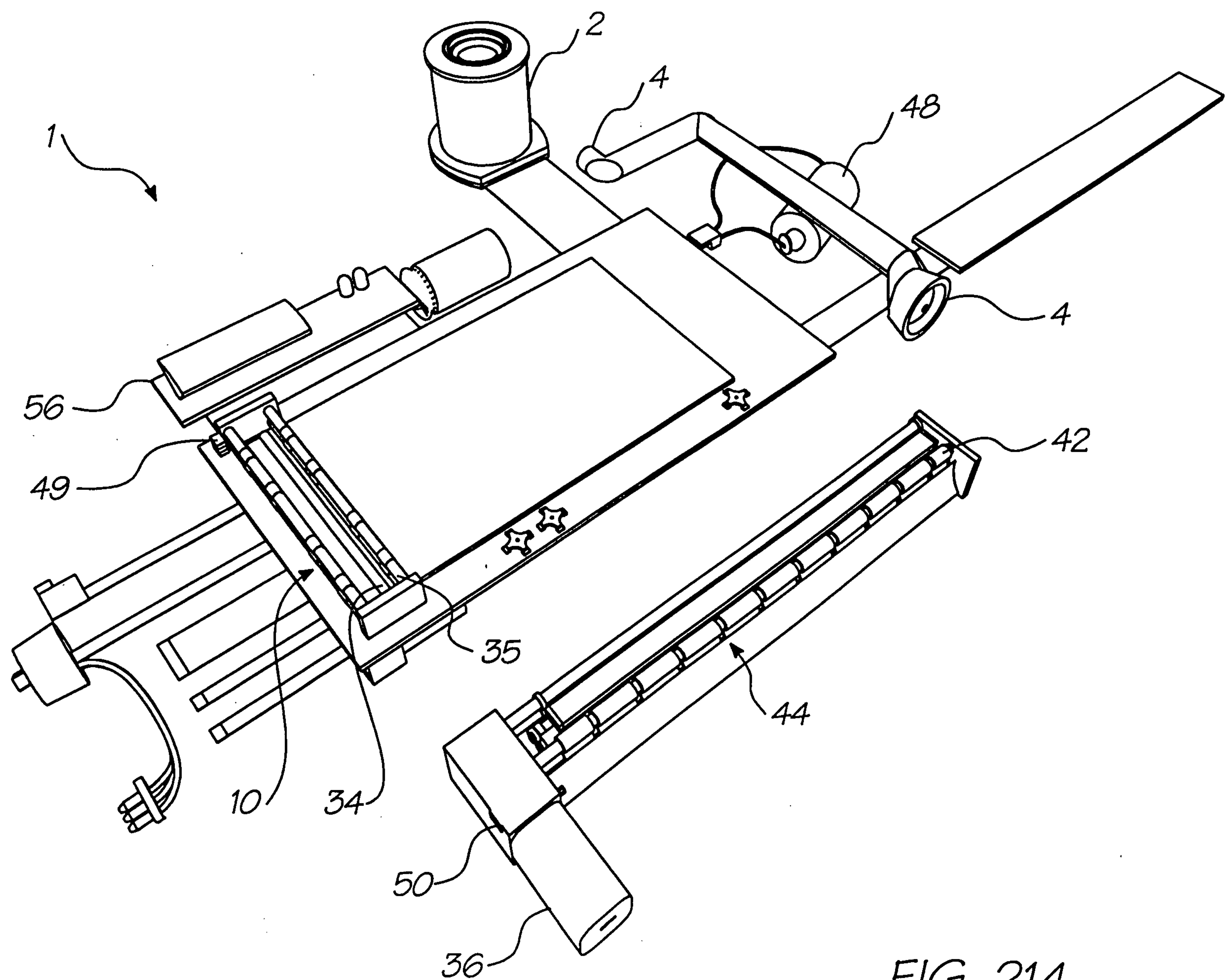


FIG. 214

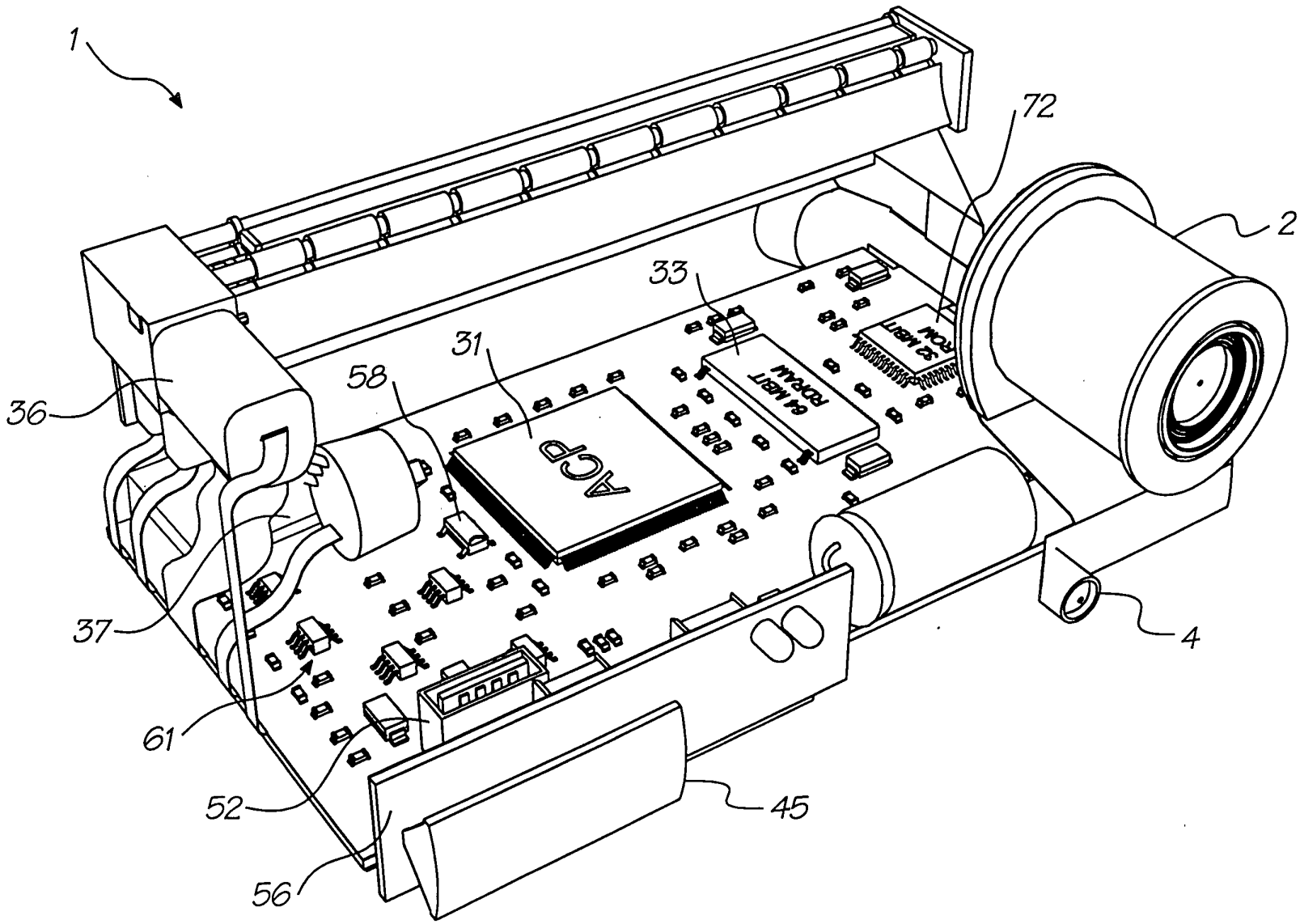


FIG. 215

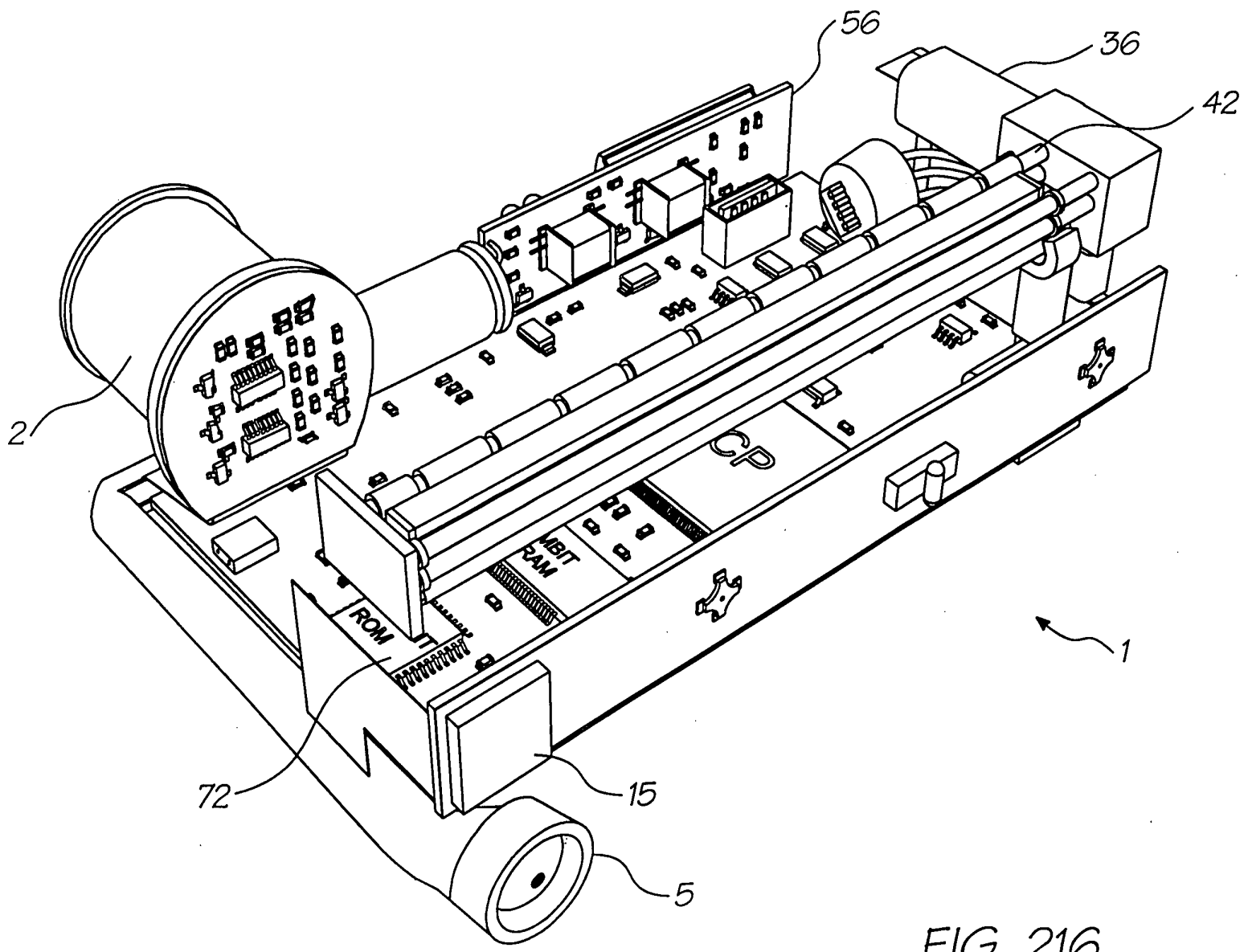


FIG. 216

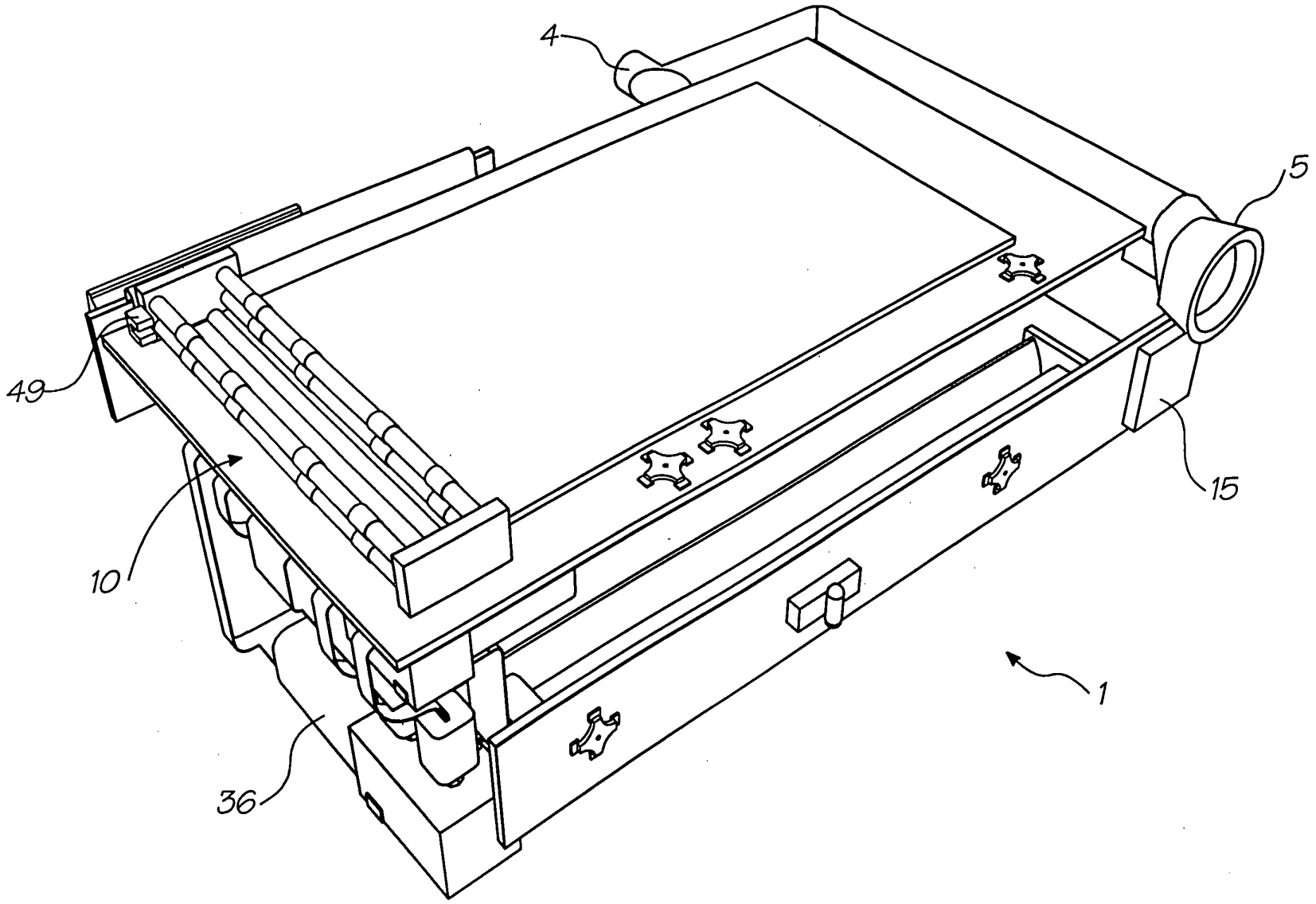


FIG. 217

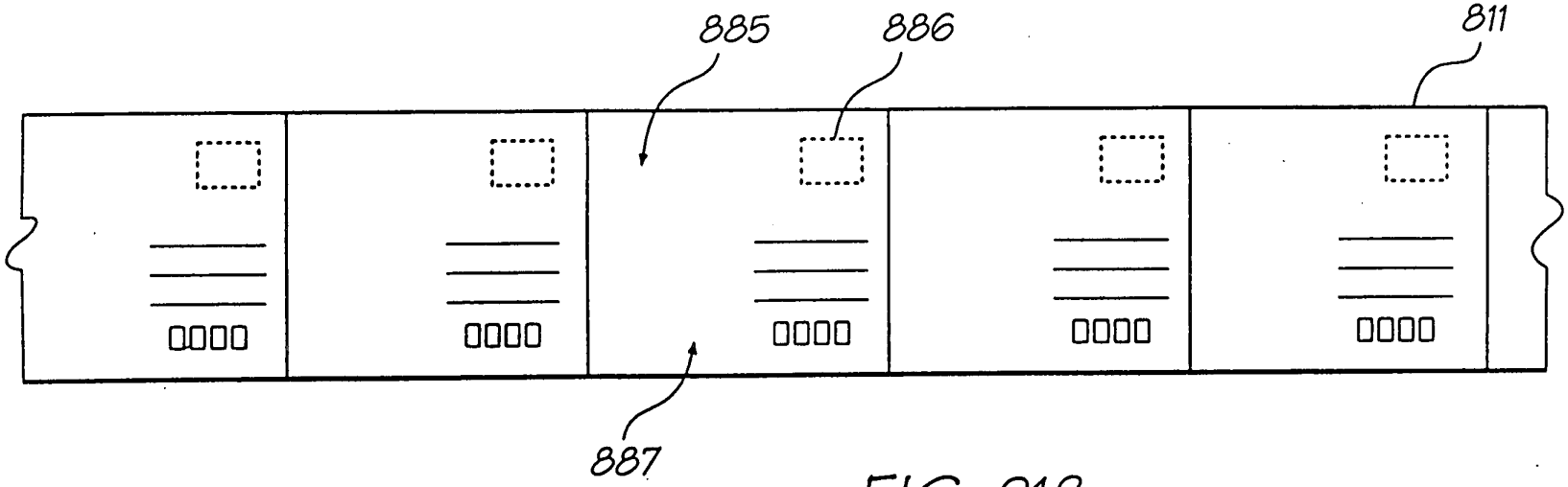


FIG. 218

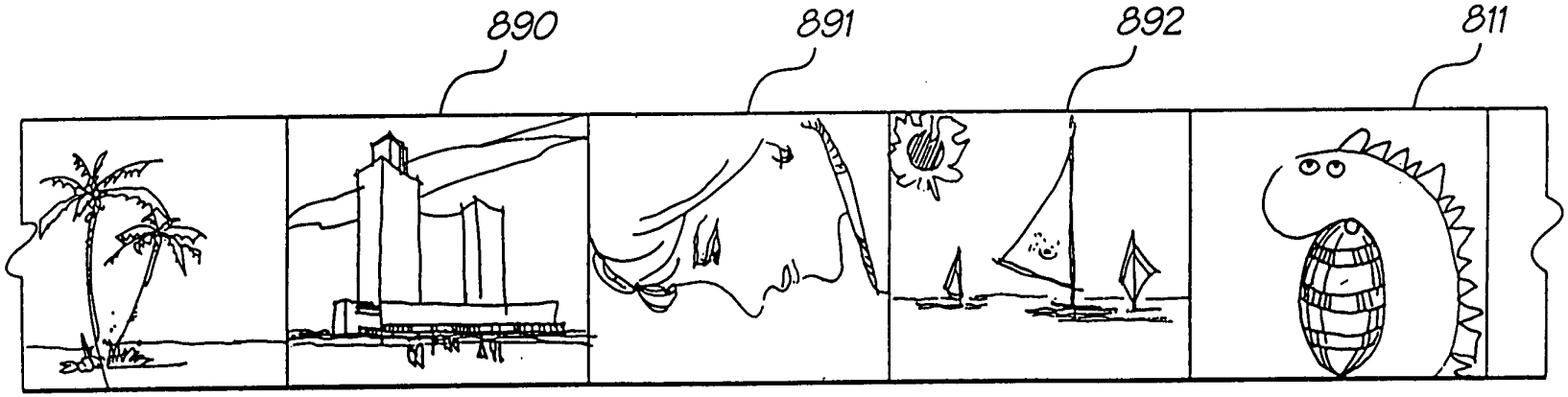


FIG. 219

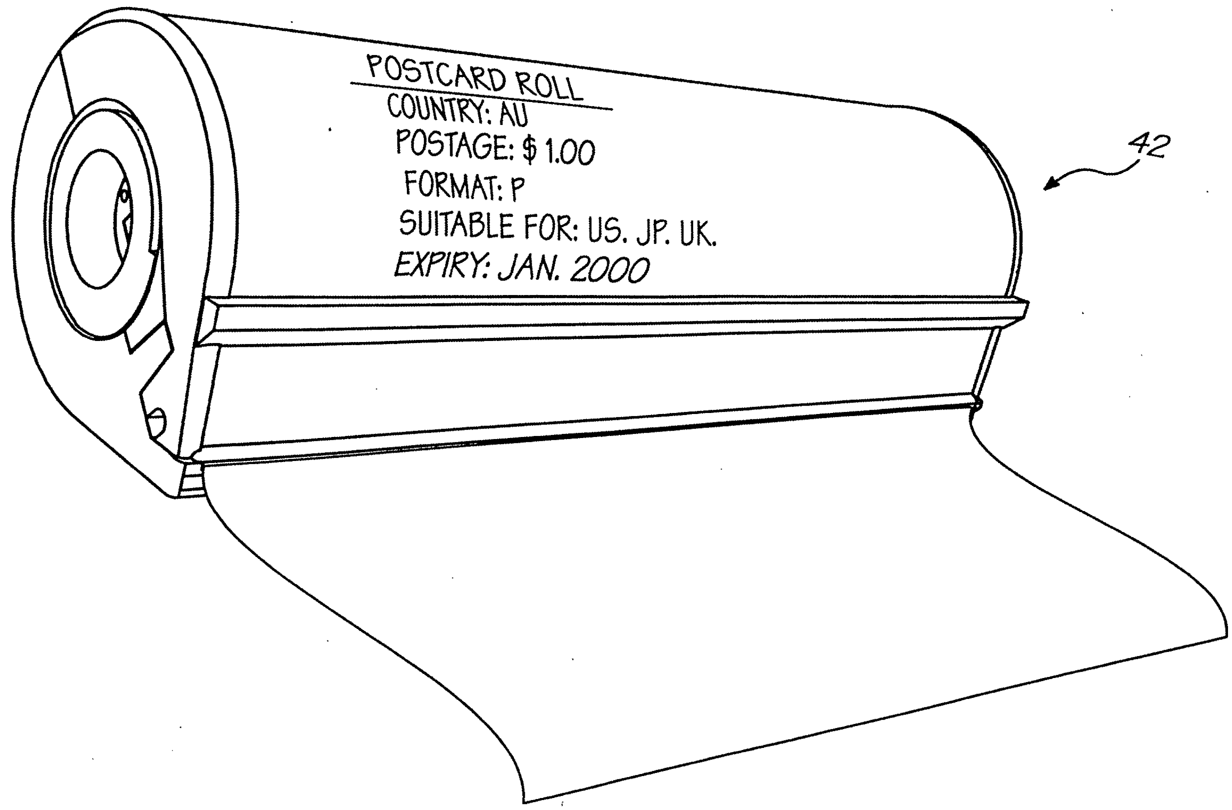

POSTCARD ROLL
COUNTRY: AU
POSTAGE: $ 1.00
FORMAT: P
SUITABLE FOR: US. JP. UK.
EXPIRY: JAN. 2000
42

FIG. 220

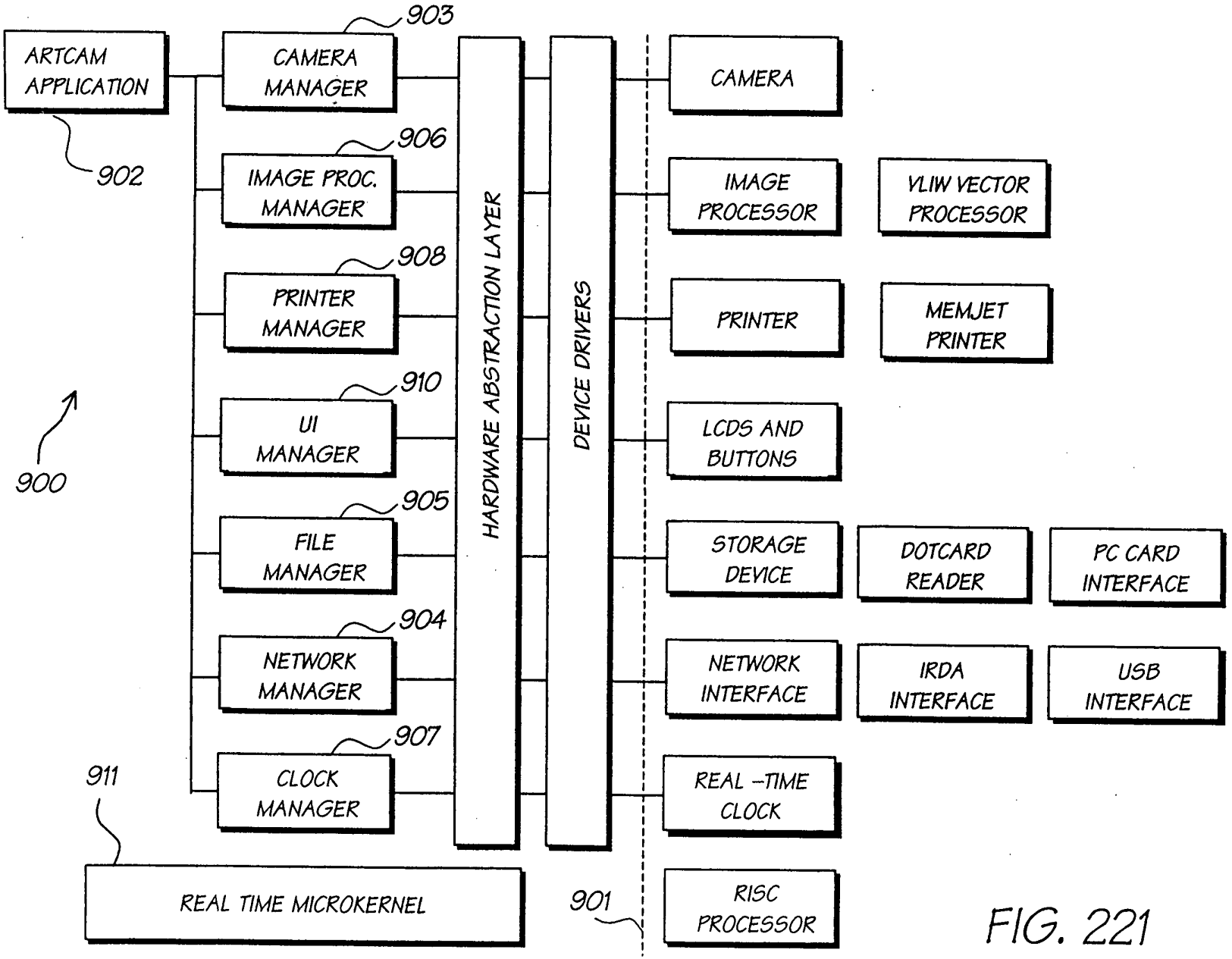

ARTCAM APPLICATION
902
900
CAMERA MANAGER
903
IMAGE PROC. MANAGER
906
PRINTER MANAGER
908
UI MANAGER
910
FILE MANAGER
905
NETWORK MANAGER
904
CLOCK MANAGER
907
911
REAL TIME MICROKERNEL
HARDWARE ABSTRACTION LAYER
DEVICE DRIVERS
CAMERA
IMAGE PROCESSOR
VLIW VECTOR PROCESSOR
PRINTER
MEMJET PRINTER
LCDS AND BUTTONS
STORAGE DEVICE
DOTCARD READER
PC CARD INTERFACE
NETWORK INTERFACE
IRDA INTERFACE
USB INTERFACE
REAL -TIME CLOCK
901
RISC PROCESSOR

FIG. 221

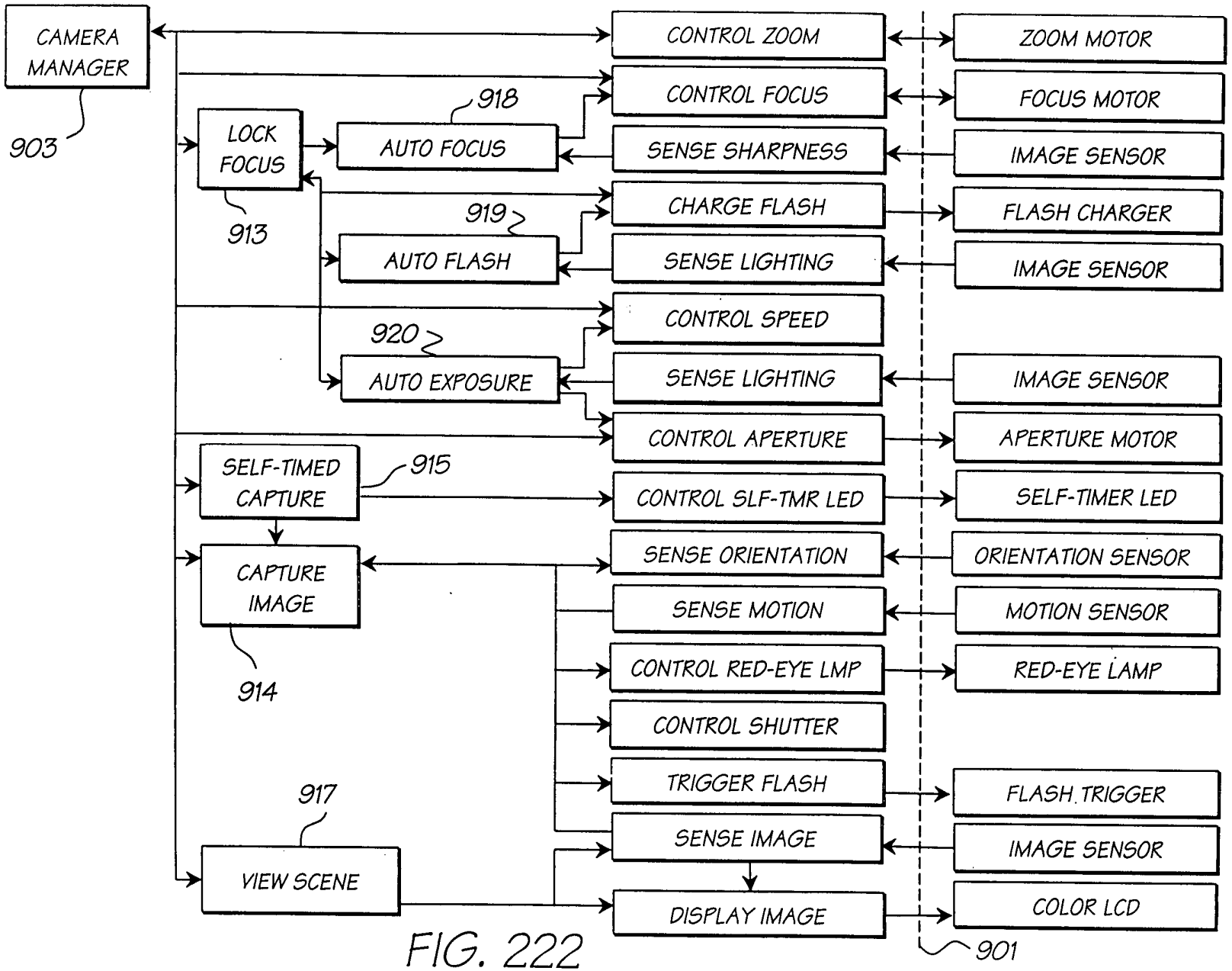

CAMERA MANAGER
903
LOCK FOCUS
913
AUTO FOCUS
918
AUTO FLASH
919
AUTO EXPOSURE
920
SELF-TIMED CAPTURE
915
CAPTURE IMAGE
914
VIEW SCENE
917
CONTROL ZOOM
CONTROL FOCUS
SENSE SHARPNESS
CHARGE FLASH
SENSE LIGHTING
CONTROL SPEED
SENSE LIGHTING
CONTROL APERTURE
CONTROL SLF-TMR LED
SENSE ORIENTATION
SENSE MOTION
CONTROL RED-EYE LMP
CONTROL SHUTTER
TRIGGER FLASH
SENSE IMAGE
DISPLAY IMAGE
ZOOM MOTOR
FOCUS MOTOR
IMAGE SENSOR
FLASH CHARGER
IMAGE SENSOR
IMAGE SENSOR
APERTURE MOTOR
SELF-TIMER LED
ORIENTATION SENSOR
MOTION SENSOR
RED-EYE LAMP
FLASH TRIGGER
IMAGE SENSOR
COLOR LCD
901

FIG. 222

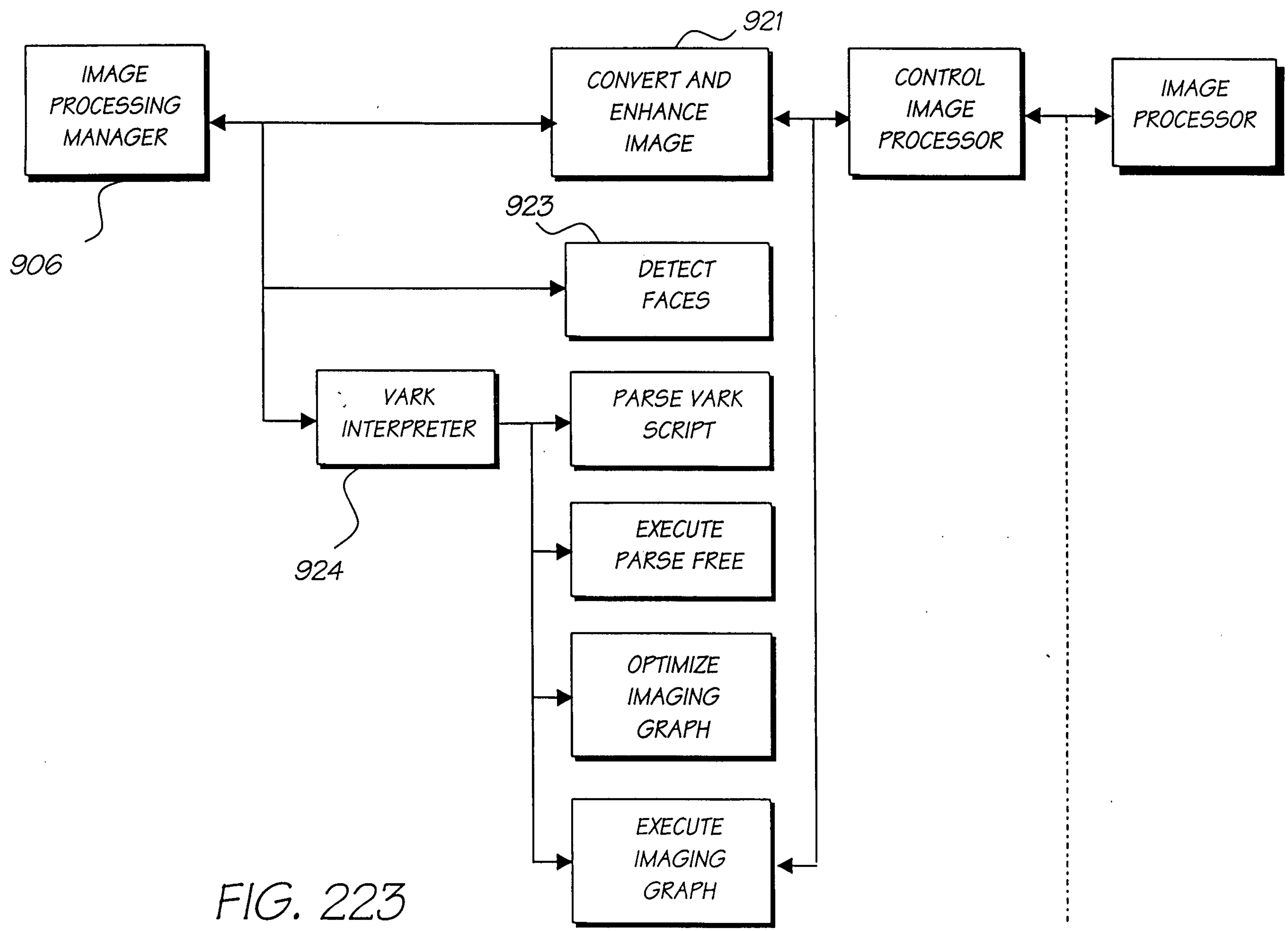


IMAGE PROCESSING MANAGER
906
CONVERT AND ENHANCE IMAGE
921
CONTROL IMAGE PROCESSOR
IMAGE PROCESSOR
923
DETECT FACES
VARK INTERPRETER
924
PARSE VARK SCRIPT
EXECUTE PARSE FREE
OPTIMIZE IMAGING GRAPH
EXECUTE IMAGING GRAPH

FIG. 223

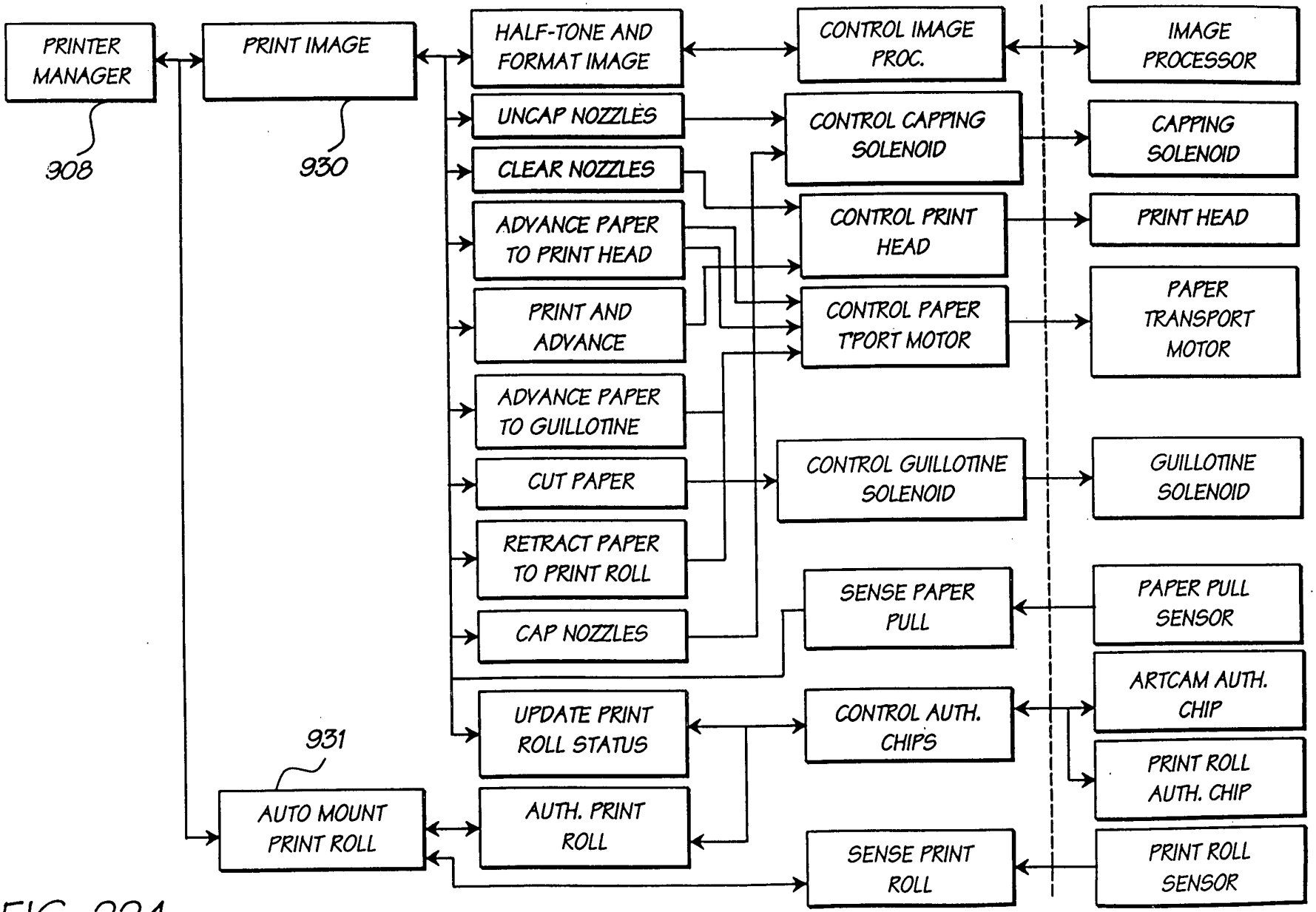

PRINTER MANAGER
908
PRINT IMAGE
930
HALF-TONE AND FORMAT IMAGE
UNCAP NOZZLES
CLEAR NOZZLES
ADVANCE PAPER TO PRINT HEAD
PRINT AND ADVANCE
ADVANCE PAPER TO GUILLOTINE
CUT PAPER
RETRACT PAPER TO PRINT ROLL
CAP NOZZLES
UPDATE PRINT ROLL STATUS
931
AUTO MOUNT PRINT ROLL
AUTH. PRINT ROLL
CONTROL IMAGE PROC.
CONTROL CAPPING SOLENOID
CONTROL PRINT HEAD
CONTROL PAPER T'PORT MOTOR
CONTROL GUILLOTINE SOLENOID
SENSE PAPER PULL
CONTROL AUTH. CHIPS
SENSE PRINT ROLL
IMAGE PROCESSOR
CAPPING SOLENOID
PRINT HEAD
PAPER TRANSPORT MOTOR
GUILLOTINE SOLENOID
PAPER PULL SENSOR
ARTCAM AUTH. CHIP
PRINT ROLL AUTH. CHIP
PRINT ROLL SENSOR

FIG. 224

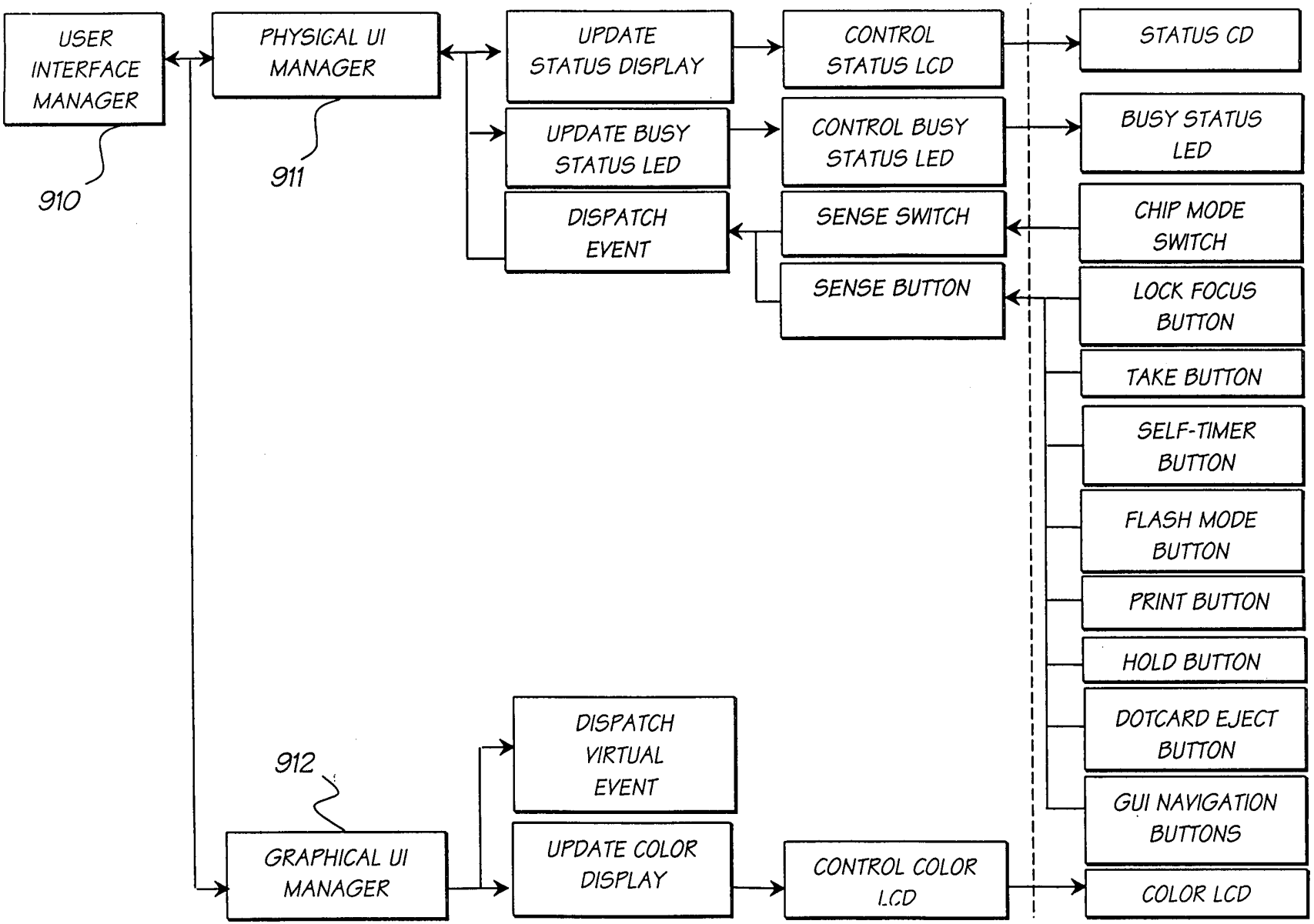

USER INTERFACE MANAGER
910
PHYSICAL UI MANAGER
911
UPDATE STATUS DISPLAY
UPDATE BUSY STATUS LED
DISPATCH EVENT
CONTROL STATUS LCD
CONTROL BUSY STATUS LED
SENSE SWITCH
SENSE BUTTON
STATUS CD
BUSY STATUS LED
CHIP MODE SWITCH
LOCK FOCUS BUTTON
TAKE BUTTON
SELF-TIMER BUTTON
FLASH MODE BUTTON
PRINT BUTTON
HOLD BUTTON
DOTCARD EJECT BUTTON
GUI NAVIGATION BUTTONS
912
GRAPHICAL UI MANAGER
DISPATCH VIRTUAL EVENT
UPDATE COLOR DISPLAY
CONTROL COLOR LCD
COLOR LCD

FIG. 225

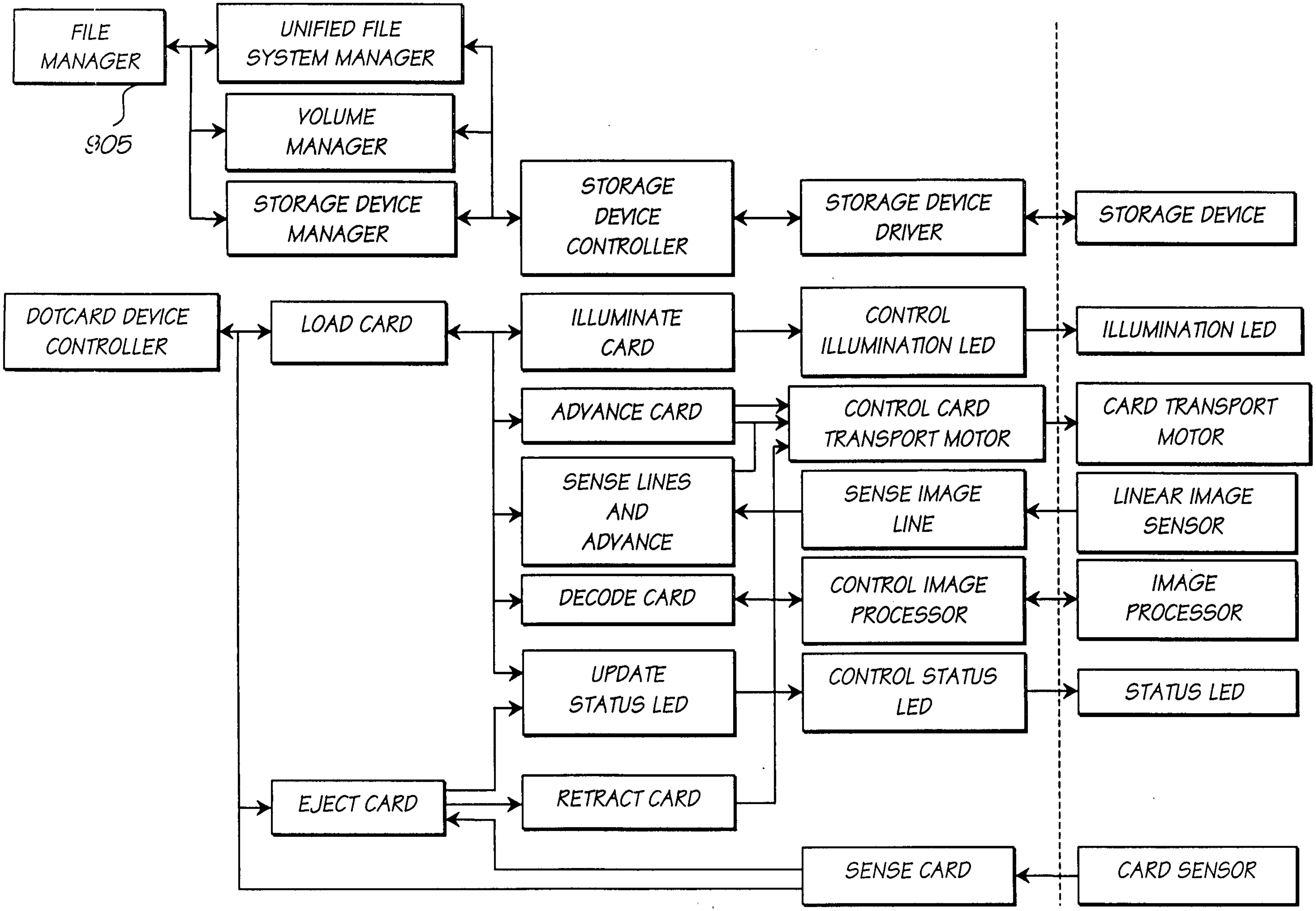

FILE MANAGER
905
UNIFIED FILE SYSTEM MANAGER
VOLUME MANAGER
STORAGE DEVICE MANAGER
STORAGE DEVICE CONTROLLER
STORAGE DEVICE DRIVER
STORAGE DEVICE
DOTCARD DEVICE CONTROLLER
LOAD CARD
ILLUMINATE CARD
CONTROL ILLUMINATION LED
ILLUMINATION LED
ADVANCE CARD
CONTROL CARD TRANSPORT MOTOR
CARD TRANSPORT MOTOR
SENSE LINES AND ADVANCE
SENSE IMAGE LINE
LINEAR IMAGE SENSOR
DECODE CARD
CONTROL IMAGE PROCESSOR
IMAGE PROCESSOR
UPDATE STATUS LED
CONTROL STATUS LED
STATUS LED
EJECT CARD
RETRACT CARD
SENSE CARD
CARD SENSOR

FIG. 226

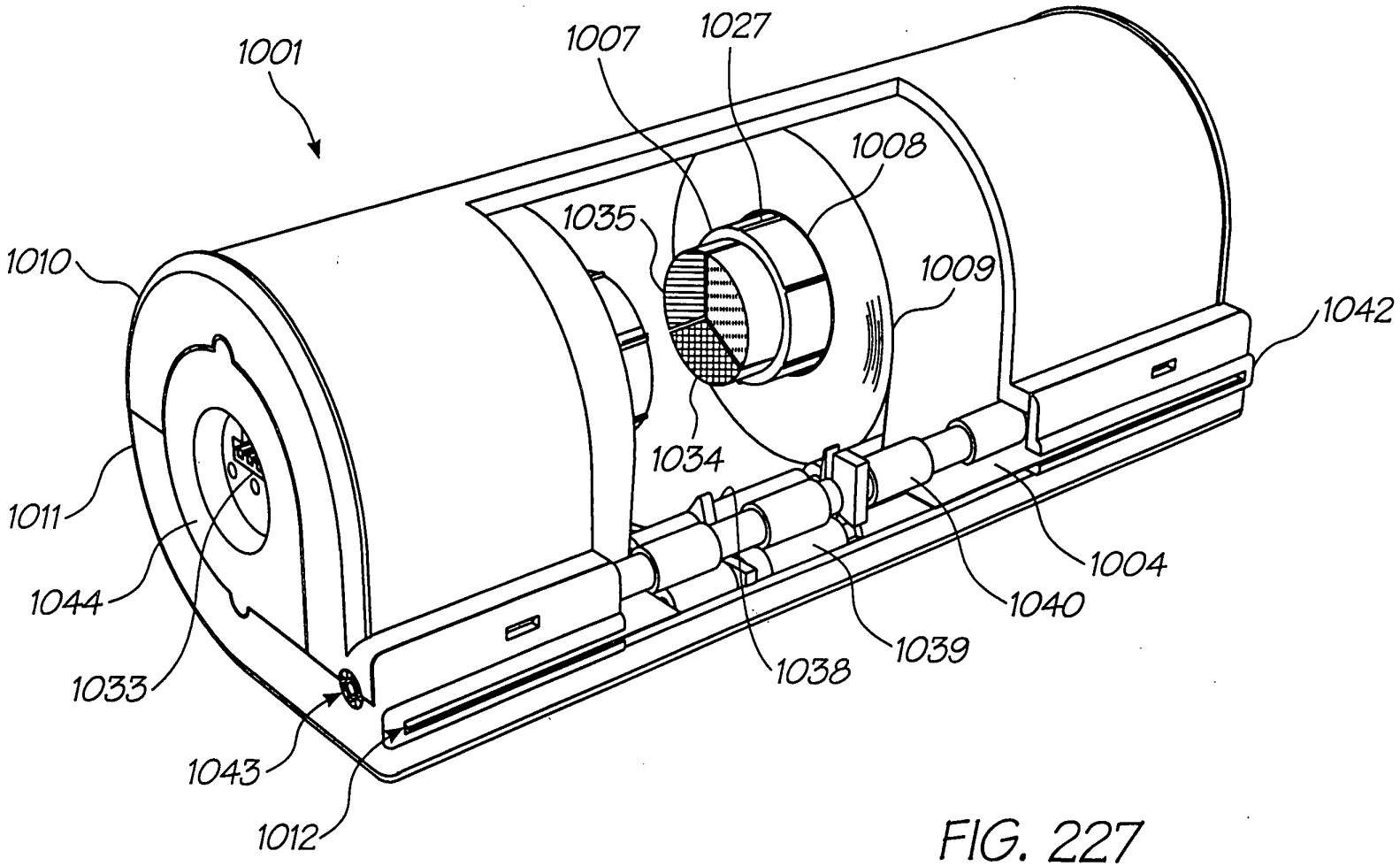

1001
1007
1027
1008
1035
1009
1010
1042
1034
1011
1004
1044
1040
1039
1033
1038
1043
1012

FIG. 227

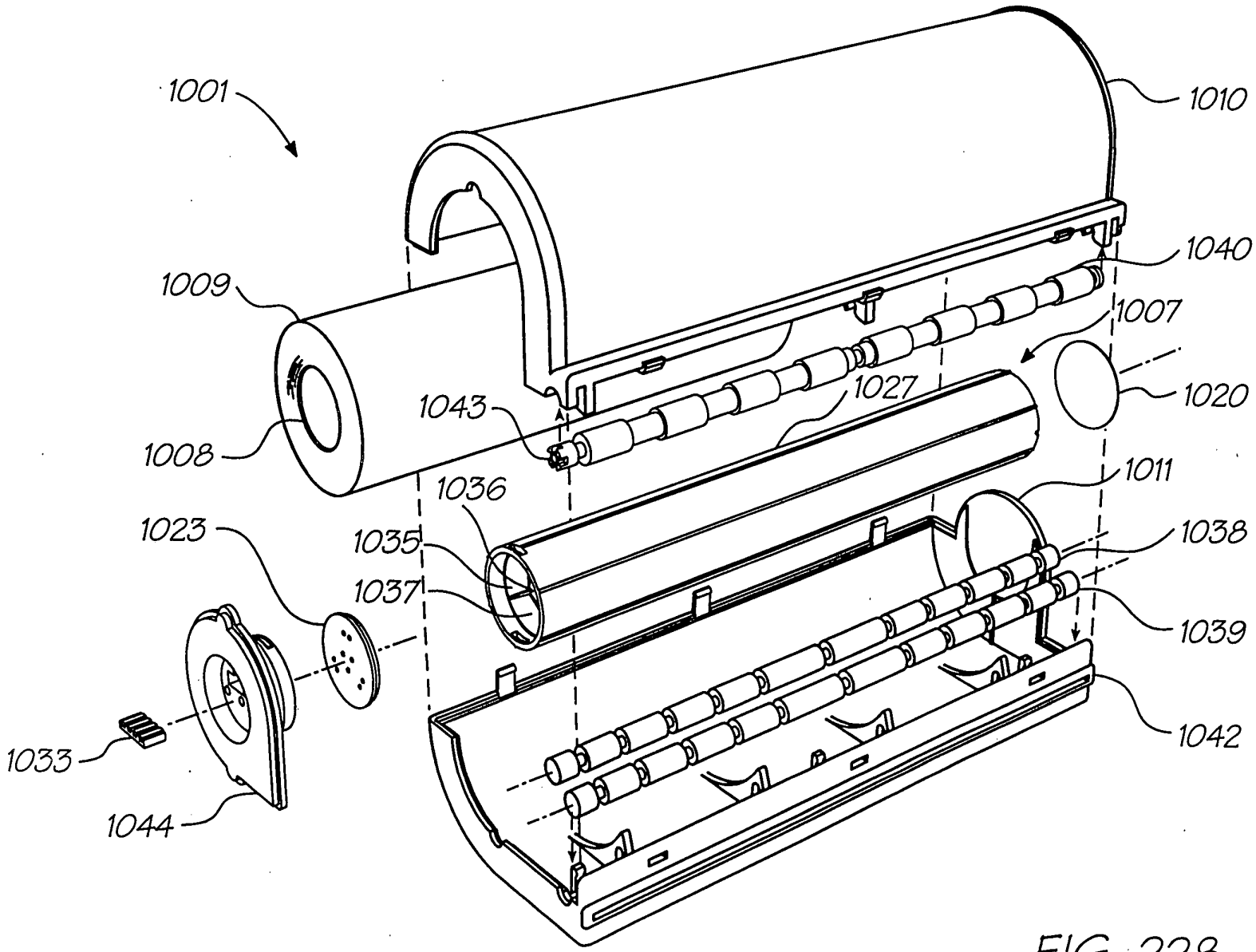

1001
1010
1040
1009
1007
1020
1027
1043
1008
1011
1036
1023
1035
1038
1037
1039
1033
1042
1044

FIG. 228

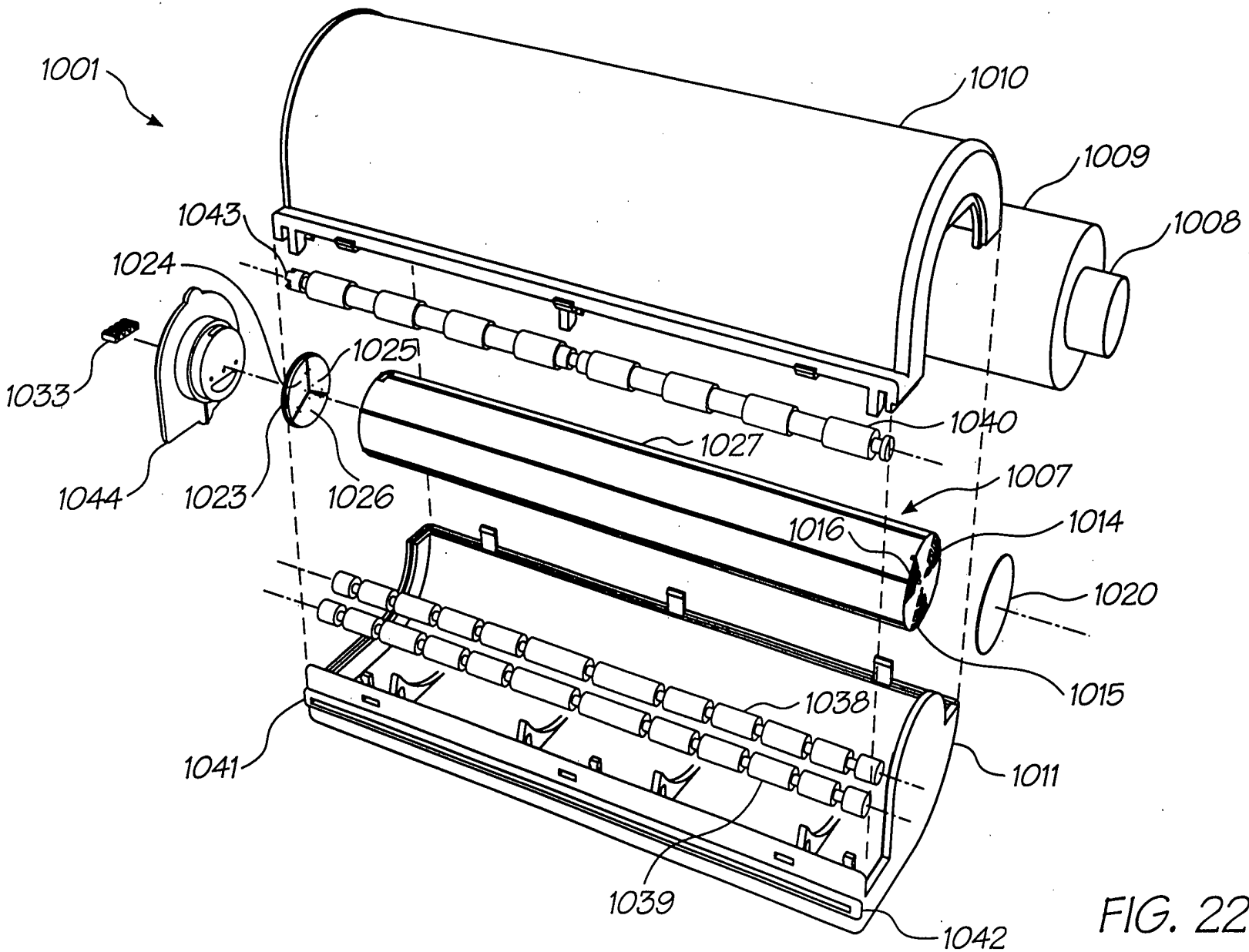

1001
1010
1009
1008
1043
1024
1033
1025
1040
1027
1007
1044
1023
1026
1016
1014
1020
1015
1038
1041
1011
1039
1042

FIG. 229

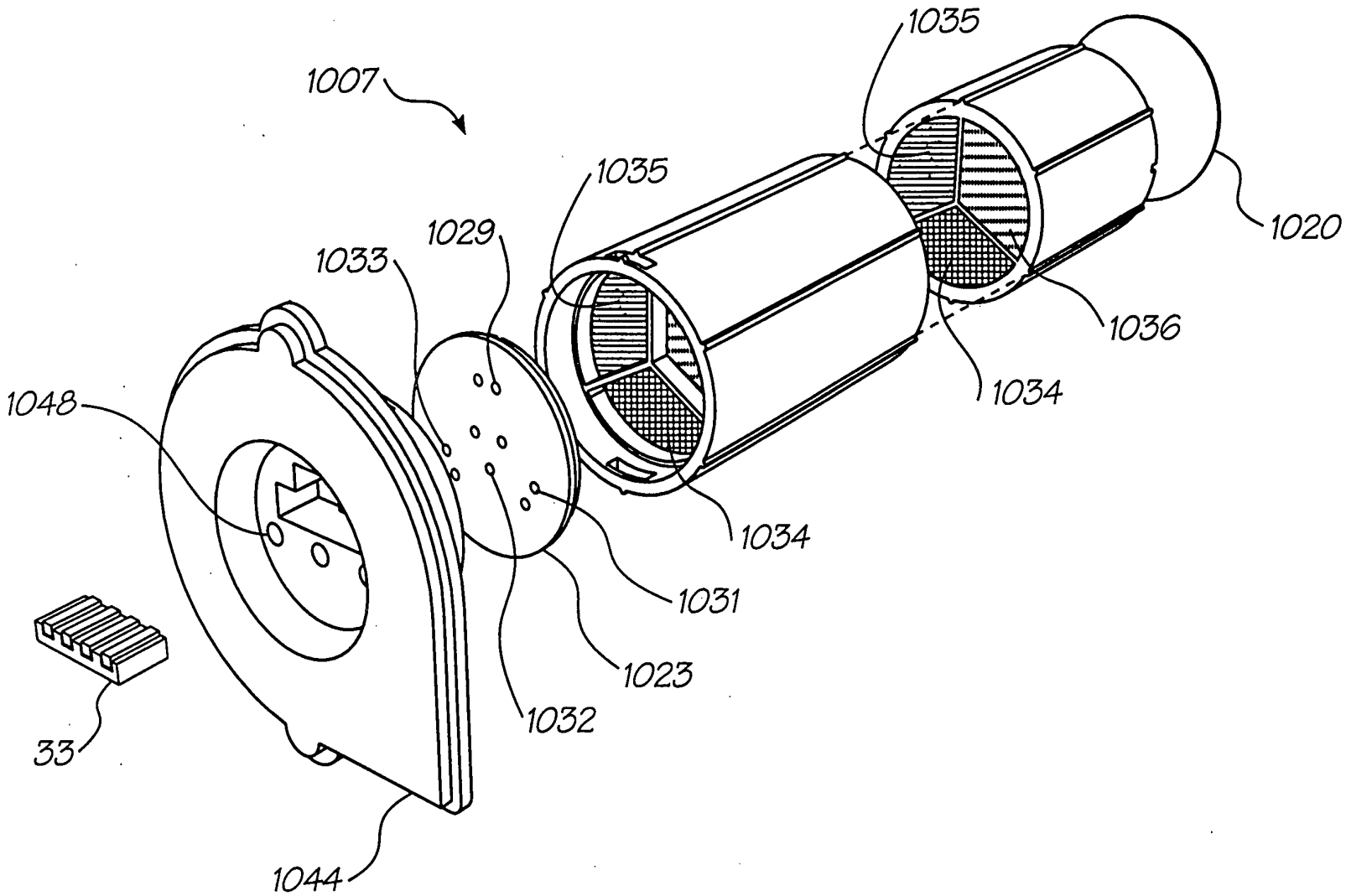

1007
1035
1035
1020
1029
1033
1036
1048
1034
1034
1031
1023
1032
33
1044

FIG. 230

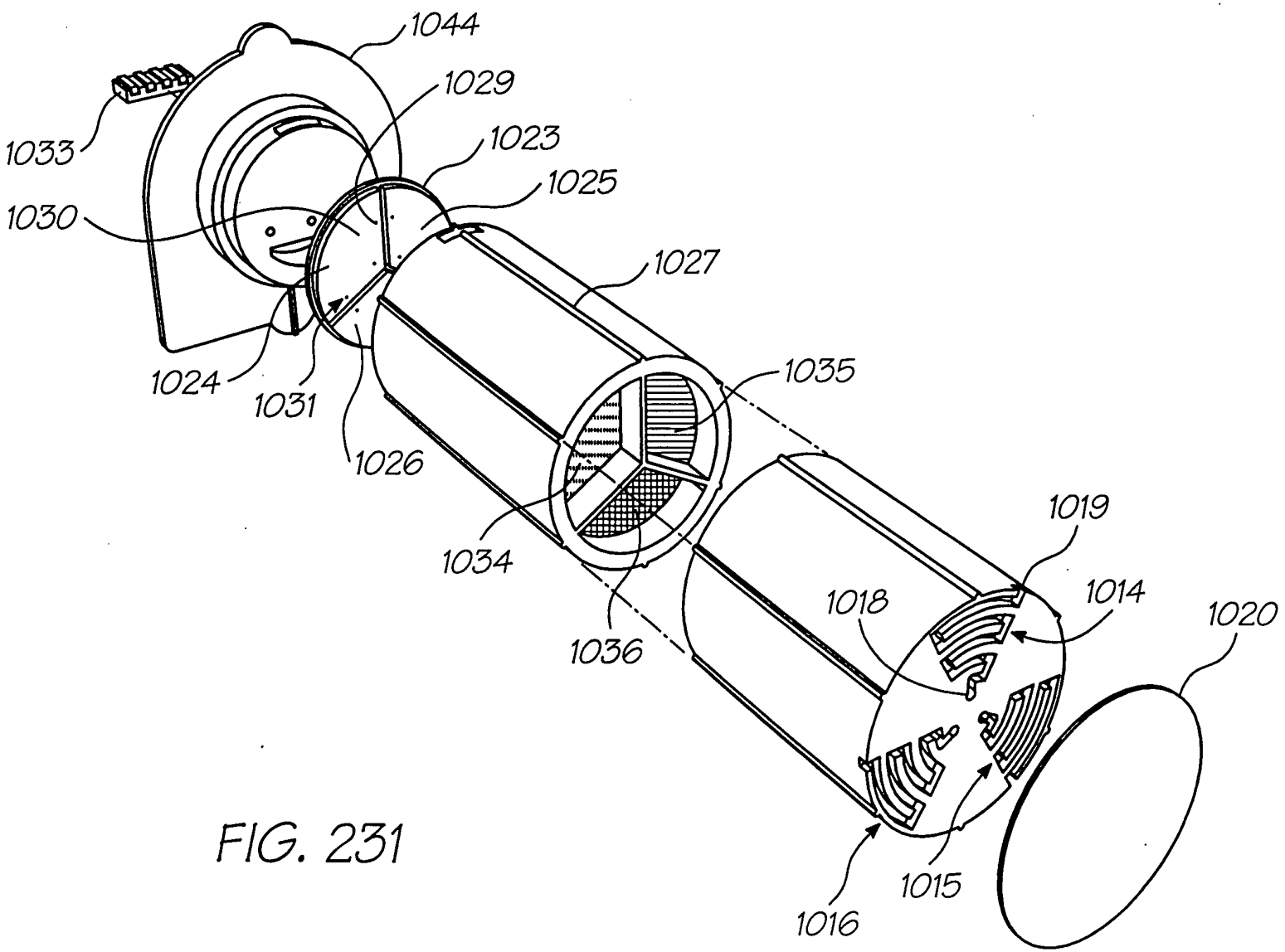

1044
1029
1033
1023
1025
1030
1027
1035
1024
1031
1026
1034
1019
1018
1014
1020
1036
1015
1016

FIG. 231